DESIGN AND ANALYSIS OF AUTO SCALING PULSED ANALOG NEURAL CIRCUITS

DIPANKAR BHATTACHARYA, B. Tech. (Hons.)

DESIGN AND ANALYSIS OF AUTO SCALING PULSED ANALOG NEURAL CIRCUITS

By

©Dipankar Bhattacharya, B.Tech (Hons.)

A thesis

submitted to the School of Graduate Studies

in partial fulfillment of the requirements for

the degree of Master of Engineering

Faculty of Engineering and Applied Sciences

Memorial University of Newfoundland

St. John's, Newfoundland, Canada A1B 3X5

August, 1991

ISBN 0-315-73347-0

Abstract

Minimization of synaptic area is important in a neural network with a high synapse to neuron ratio. Consequently one has to optimize the synapse rather than the neuron. A pulsed analog network with amplitude modulation results in a very compact and efficient synapse. Charge summation is used which leads to a single bus as the summer. Membrane capacitance has been distributed to the synapses allowing the network to be perfectly scaled. Like the biological neuron, the neuron fires a single output pulse when the activation exceeds the threshold. A discharge pulse is generated to discharge the membrane capacitances via discharge transistors which have also been distributed to synapses for scaling purposes. Circuit design and detailed analysis has been included along with simulation results. Standard cells have also been presented. As the proposed architecture behaves quite differently from existing architectures, simulation of some of the standard examples of neural networks have been included. Two chips have also been designed using 3μm design rules.

Acknowledgement

I sincerely acknowledge and thank my supervisor Prof. Bruce-Lockhart for all his help, useful discussions, criticisms and encouragement for my work for the whole duration of my program here. I thank The School of Graduate Studies of Memorial University, the Faculty of Engineering, the Associate Dean of Engineering (Graduate Studies) and his office for the necessary financial support which I needed badly. I also thank all the staff members of C-CAE and specially to Lloyd Little for making my thesis more presentable, at least in terms of the figures. I also thank all my fellow graduate students in the Faculty of Engineering. Finally, I deeply acknowledge the constant encouragement of my wife and our families.

Contents

Abstract ii

Acknowledgement iii

Contents iv

List of Figures vii

List of Tables x

List of Symbols xi

1 Introduction 1

2 Neurons and Neural Networks 4

2.1 Introduction 4

2.2 Neuron 4

2.3 Neural Network 7

2.3.1 General Review 7

2.3.2 Learning 11

2.4 Concluding remarks 15

3 Literature Review 16

3.1 Introduction 16

3.2 Review 17
3.2.1 Analog implementation 17
3.2.2 Digital implementation 23
3.3 Concluding remarks 25

4 Design Philosophy & Proposed Architecture **26**
4.1 Introduction 26
4.2 Objectives 26
4.2.1 Motivations 27
4.3 Proposed Architecture 28
4.4 Concluding remarks 32

5 Circuit Design and Analysis **33**
5.1 Introduction 33
5.2 Excitatory Synapse 33
5.2.1 Circuit Description 33
5.2.2 Circuit Design 36
5.2.3 Circuit Analysis 41
5.3 Inhibitory Synapse 43
5.3.1 Circuit Description 43
5.3.2 Circuit Design 46
5.3.3 Circuit Analysis 46
5.4 Standard Neuron 48
5.4.1 Circuit Description 48
5.4.2 Circuit Design 48
5.4.3 Circuit Analysis 51
5.5 Input Neurons 53

5.5.1 Circuit Description 53
5.5.2 Design & Analysis of standard input neuron 56
5.5.3 Design & Analysis of Inverting Input Neuron 57
5.6 Concluding remarks 59

6 Standard Cells **62**
6.1 Introduction 62
6.2 Cell Specifications 63
6.3 Cell Description 65
6.3.1 Excitatory Synapse 66
6.3.2 Inhibitory Synapse 67
6.3.3 Standard Neuron 68
6.3.4 Inverting input neuron 76
6.3.5 Standard input neuron 79
6.4 Simulation 82

7 Simulations and Results **85**
7.1 Pattern Classifier 85
7.2 XOR Gate 95
7.3 Cooperative Assignments 106
7.4 Implementation 107
7.5 Concluding Remarks 107

8 Conclusions **113**

References **115**

Appendix **121**

List of Figures

2.1 Structure of a classical neuron (adapted from [Mead, 89]). 6

2.2 Block diagram of a typical artificial neuron. 8

2.3 A typical multi-layered feedforward network. 13

4.1 Block diagram of the proposed neural architecture 29

5.1 Schematic of the excitatory synapse 34

5.2 Normalized firing rate of the neuron without leakage. 37

5.3 Activation curves for different values of τ (when leakage is included). 39

5.4 Simulation result of a single synapse 42

5.5 Activation voltage generated using equations 5.13, 5.16 and 5.17 . 44

5.6 Schematic diagram of the inhibitory synapse. 45

5.7 Spice simulation of three excitatory and one inhibitory synapses. . 47

5.8 Schematic of the standard neuron. 49

5.9 Spice simulation of the standard neuron. 50

5.10 Schematic diagram of the standard input neuron. 54

5.11 Schematic diagram of the inverting input neuron. 55

5.12 Hspice simulation of the standard input neuron. 58

5.13 Hspice simulation of the inverting input neuron. 60

6.1 Layout of the excitatory synapse 66

6.2 Layout of the inhibitory synapse 67

6.3 Layout of the standard neuron 69
6.4 Layout of the comparator. 70
6.5 Layout of the buffer 71
6.6 Layout of the inverter3. 72
6.7 Layout of the inverter2. 73
6.8 Layout of the two input NAND gate. 74
6.9 Layout of the inverter. 75
6.10 Layout of ramp generator inN0. 77
6.11 Layout of the inverting input neuron. 78
6.12 Layout of inN1. 80
6.13 Layout of the standard input neuron. 81
6.14 Simulation of the extracted layout of the excitatory synapse. . . . 83
6.15 Simulation on the extracted schematic of the standard neuron. . . 84

7.1 Schematic of the template matching example. 86
7.2 Spice simulation of the template matching example. 87
7.3 Spice simulation of the template matching example. 88
7.4 Spice simulation of the template matching example. 90
7.5 Spice simulation of the template matching example. 91
7.6 Content addressable memory. 93
7.7 Output of 7 neurons when presented with 01100. 94
7.8 Schematic of the first XOR circuit. 97
7.9 Plots for input 00. 98
7.10 Plots for input 01. 99
7.11 Schematic of the second XOR circuit. 100
7.12 Plots for input 01. 101
7.13 Plots for input 11. 102

7.14 Schematic of the third XOR circuit. 103
7.15 Plots for input 01. 104
7.16 Plots for input 11. 105
7.17 3x3 cooperative assignment network. 108
7.18 Output of all 9 neurons of the 3x3 cooperative assignment network. 109
7.19 Schematic of controllable pattern classifier/CAM. 110
7.20 Hspice simulation of the CAM. 111
7.21 Layout of the network obtained by auto place and route routines. 112

List of Tables

5.1 Control voltages and the periods of the generated pulses for the standard input neuron. 57

5.2 Control voltages and the periods of the generated pulses for the inverting input neuron. 59

7.1 Weight distribution of the 3x3 cooperative assignment net. 106

List of Symbols

Symbol	Description
$\beta = K' \frac{W}{L}$	- transconductance parameter
CCD	- charge coupled device
CJ	- zero bias junction bottom capacitance density of the moat bulk diffusion
$CJSW$	- zero bias sidewall capacitance density
F	- Faraday constant
γ	- bulk threshold parameter
K'_p	- transconductance parameter for PMOS
K'_n	- transconductance parameter for NMOS
K'	- transconductance parameter
L	- length of transistor
λ	- channel length modulation parameter
MJ	- bulk junction bottom grading coefficient
$MJSW$	- bulk junction sidewall grading coefficient
$MNOS$	- metal nitride oxide semiconductor
ϕ	- surface potential
ϕ_B	- built in potential
R	- universal gas constant
V_m	- membrane voltage
V_t	- threshold voltage of MOS transistor

V_{T0}	- zero bias threshold voltage of MOS transistor
V_{wt}	- weight voltage
W_{ij}	- weight from neuron j to neuron i
W	- width of a transistor
$S = \frac{W}{L}$	- shape factor
Z	- valency

Chapter 1

Introduction

Research concerning neural networks can be traced back a few decades, but it has been mostly on the theoretical studies and computer simulation. Computer simulation is slow and the real power of the neural network can best be extracted only when one goes for specialized circuits in microelectronics. In the last ten years or so, a lot of literature has been published on neural circuitry and their implementation in silicon (chapter 3). To date, however, most of the networks reported are small in scale. The computing power of neural networks lies in their connections. In biological systems, one neuron may be connected to thousands of other neurons. So, one has to consider the implications of the scale if one is ever to approach the size of networks present in our biological system.

One neuron is connected to another neuron through a synapse. If there are N neurons in a network, then the number of synapses grows as N^2 for a fully connected network (such as a Hopfield net). From a circuit point of view, it is not all that easy to connect a large number of synapses together and feed the output to a neuron. However, for an auto scaling circuit, the number of synapses per neuron is not limited by any circuit constraint.

An auto scaling pulsed neural network is presented in this thesis. It leads to a very compact synapse which is highly desirable in a network where synapses

outnumber neurons. It further enables one to add the outputs of a large number of synapses together. The auto scaling feature has also enabled us to design standard cells which can be plugged together to realize networks of varying sizes.

Pulse-stream analog networks have already been reported in the literature (chapter 3). Under this scheme, the neural state is represented by pulses whose frequency depends on the input activation. But the proposed circuits differ in many respects. The scalability has been achieved by distributing the membrane capacitances in the excitatory synapses. Synapses can be either excitatory or inhibitory but cannot switch back and forth between excitation and inhibition. Each time the activation voltage goes past the threshold, an output pulse is generated. At the same time one discharge pulse is also generated to discharge the membrane capacitances (like repolarization in biological neurons) so that the charge integration cycle starts all over again. For the scaling purpose, discharge transistors are also distributed in synapses.

These neural circuits have been designed, and simulated using the Spice program. Detailed mathematical analysis has also been done. A number of standard networks like pattern classifier, content addressable memory, XOR gates, Hopfield nets etc. have been simulated to examine proper operation of the designed circuits. A standard cell library has also been developed.

The thesis has been organized as follows. The second chapter introduces the biological neurons followed by artificial neural networks. Two learning schemes have also been included there. The next chapter deals with the literature review - how different aspects of neural networks have been achieved by different people. The fourth chapter gives the design philosophy. The proposed neural architecture is also presented there. The fifth chapter gives the circuit design and analysis. Mathematical equations are also presented which can be used for developing a fast

simulator. Chapter six deals with the standard cells including the cell design philosophy along with layout of several standard cells. The seventh chapter describes simulation results of different neural networks. It also includes the schematic diagram and layout of one of the two chips that would be fabricated. Finally, in the last chapter I conclude my present work and suggest some areas where further work can be done.

Chapter 2

Neurons and Neural Networks

2.1 Introduction

This chapter deals with a brief introduction to biological neurons and artificial neural netwoks. A brief discussion on the neuron followed by a simple description on the generation of action potential is presented. Different aspects of neural networks including two popular learning schemes have also been included.

2.2 Neuron

The neuron is the basic anatomical unit of the nervous system. A typical neural cell (figure 2.1) has four distinct regions - cell body, dendrites, axon and the presynaptic terminals of the axon. The cell body is the source of energy for the neural information processing. It gives rise to a tubular process known as the axon which can extend over a large distance. The axon, in turn, divides into a large number of presynaptic terminals. These presynaptic terminals contact with the postsynaptic terminals (dendrites) of the other neurons at the synaptic sites. The neuron integrates the incoming signals from other connecting neurons by the capacitance of the cell body and fires an output pulse (action potential) when the total input activation exceeds some threshold voltage. Some axons are covered

with an insulating material called myelin to reduce the capacitance between the cytoplasm and the extracellular fluid. This is essential for achieving high speed conduction. The myelin sheath is interrupted at regular intervals by the nodes of Ranvier where the transmitted signals are periodically restored.

Nerve cell, like other cells, has different concentrations for different ions across its membrane [Koester, 81]. Out of the ions, Na^+ and Cl^- concentrations are lower inside whereas K^+ and organic A^- are lower outside. Due to the concentration gradient, K^+ ions tend to move out across the membrane through diffusion. This diffusion leads to separation of charges and hence a potential difference (V_m) which impedes further passage of charge. At a voltage of around -75 mV, K^+ ions reach an equilibrium when there is no net flow of K^+ ions. This equilibrium potential can be obtained by the Nernst equation :

$$E = \frac{RT}{ZF} ln \frac{[C^+]_o}{[C^+]_i} \tag{2.1}$$

where C_o^+ and C_i^+ are the concentrations of ions in the extracellular fluid and inside the cell.

Due to the presence of Na^+ ions, the cell comes to a resting potential of about -60 mV when the net influx of Na^+ ions is totally balanced by the net efflux of K^+ ions. In order to maintain the ionic gradient, a metabolically driven Na-K pump brings in a steady supply of K^+ ions while driving Na^+ ions out of the cell. If the membrane potential is increased from -60 mV to say -70 mV, the cell is hyperpolarised reducing its ability to generate an action potential and is therefore said to be inhibited; whereas, if the potential is decreased, the cell is depolarised and is said to be excited because it increases its ability to generate the action potential.

If a nerve cell is depolarised to a small extent, the charge leaks away and the action potential is never initiated. However, if the cell is depolarised to approxi-

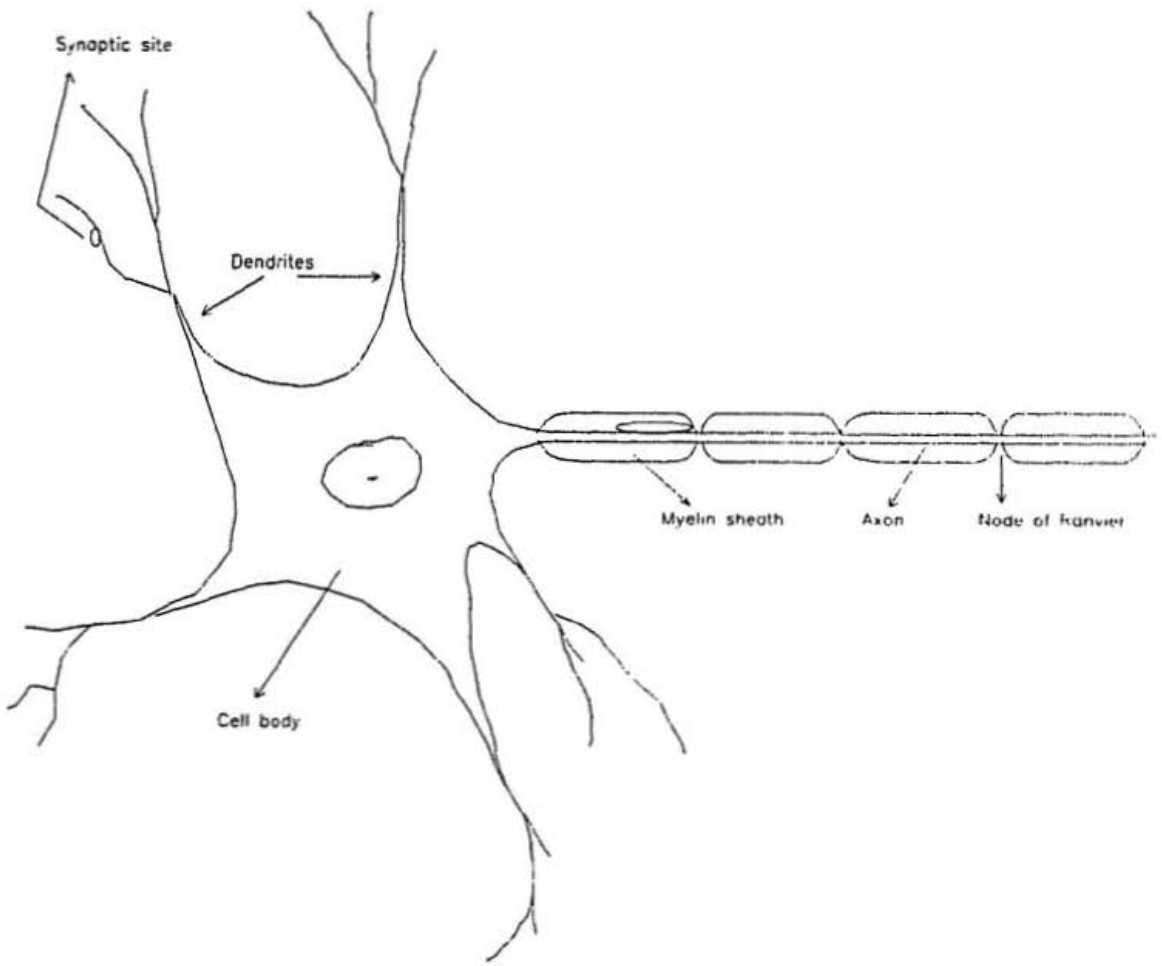

Figure 2.1: Structure of a classical neuron (adapted from [Mead, 89]).

mately -40 mV, an action potential is generated even if the voltage is brought back to -60 mV. This generation of action potential can be explained in terms of the voltage dependent ion channels [Koester, 81A]. When the cell is depolarised, Na^+ ion channels open (increasing Na^+ conductance) thereby increasing inward Na^+ current. This further depolarizes the cell which in turn opens more Na^+ channels. This regenerative process continues till the action potential is generated. However, at this stage, Na^+ conductance and hence Na^+ current starts decreasing resulting in further decrement of Na^+ conductance. At the same time, K^+ ion channels open resulting in an outward K^+ current which eventually repolarize the membrane to the resting potential.

2.3 Neural Network

2.3.1 General Review

Artificial neural networks are biologically inspired. They are networks of simple processing elements or units interconnected by weights of variable strengths. They are neural in the sense that the computation is done collectively rather than individually. In general, in a neural network, an amplifier with a non-linear output characteristic forms the cell body, wires replace axons and dendrites, and the resistors model the synaptic connections or weights among the interacting units. When a neuron is activated, it evaluates all inputs from other neurons and finds out the weighted sum. If the sum or the activation goes beyond a predetermined threshold, the neuron changes its output (figure 2.2).

In mathematical terms, if O_j represents the set of all neural outputs, then the total activation act_i of the ith neuron is

$$act_i = \sum_j W_{ij} O_j \tag{2.2}$$

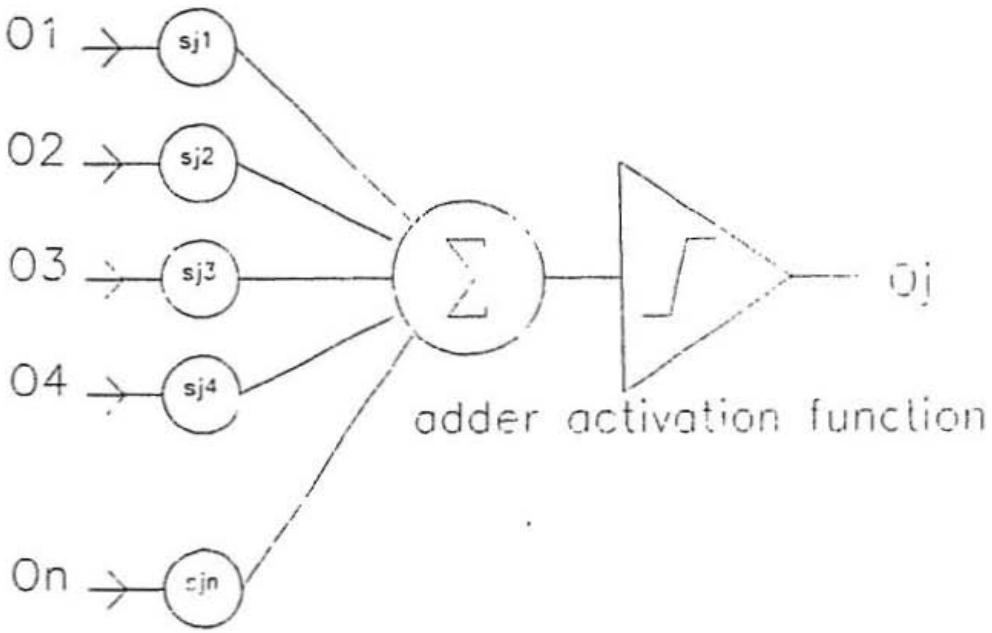

Figure 2.2: Block diagram of a typical artificial neuron.

where W_{ij} is the strength of connection from unit u_j to unit u_i. The output of u_i is given by $O_i = f(act_i, thr)$ where f is a nonlinear or decision making function and thr is the threshold voltage.

The strength of connection between two neurons determines the degree of interaction between the two. It can be either excitatory or inhibitory, normally represented by positive and negative weights. If the connection is excitatory, then the activation of the neuron is increased, while the inhibitory connection tends to reduce the activation.

When a network is activated, all the neurons operate in parallel and try to adjust their states. In the synchronous update procedure, they simultaneously update their states at each pulse of a central timing clock; while in asynchronous update, each of the neurons, at any instant of time, has a fixed probability of updating its state. Since the neurons update their states independently, in a very small timeframe only one neuron can be thought of updating its state. Whatever the updating procedure, eventually the neurons settle to a stable state representing some global configuration. This is achieved by utilizing the locally available information and the massive parallelism inherent to the system.

Different researchers have proposed networks employing different units in different configurations [Aarts et. al., 89] but most can be encompassed within the stated framework. The major differences are noted below.

- Connectivity : Connectivity varies from single layered network (e.g. Hopfield nets) to multilayered networks with hidden units (e.g. backprop nets). Backpropagation nets are also strictly feedforward and the connections are essentially unidirectional. Hopfield nets, on the other hand, have bidirectional connections. Both are discussed in more detail below.

- Neural units : Networks employing simple linear units have a set of input units and a set of output units. It can be shown that the computation done by multilayered linear units can also be done by a network without a hidden layer. The output of the simple linear model is an identity function; that is $O_i = act_i$. Kohonen has done extensive studies on this kind of networks and their learning [Aarts et. al., 89]. On the other hand, in the linear threshold unit, output $O_i = 1$ if the activation $act_i > \theta_i$ (where θ_i is the threshold) and 0 otherwise. Perceptrons are a special class of networks employing a single layer linear threshold units without any feedback. But the most common one is the one utilizing the semilinear activation function where the output $O_i = f(act_i)$, f being a monotonically non-decreasing differentiable function.

- States : Output states can either be binary; i.e., $O_i : \{0, 1\}$ in which case the function f is making a hard decision as in the perceptron model, Hopfield's content addressable memory, back propagation networks; or the output can be a continuous value in which case f is a non-linear, monotonically increasing function as in Hopfield's neural decision networks.

- Activation : The output function or the decision can also be either deterministic or probabilistic. The models employing the former are Hopfield nets, back propagation nets etc. whereas the Boltzman machine employs a probabilistic response function.

- Representation : The overall representation can be local, in which the state of individual units may represent something meaningful. On the contrary, in the distributed representation, the state of each unit has to be interpreted in conjunction with all other neurons.

It is worthwhile, in this context, to discuss Hopfield nets. In a Hopfield net,

every neuron is connected to every other neuron except for itself (i.e. $W_{ii} = 0$). The other restriction is that the weights are symmetrical, that is $W_{ij} = W_{ji}$. For a two-state neuron i, the total input is

$$x_i = \sum_j T_{ij} V_j + I_i \tag{2.3}$$

where I_i is external input to the neuron i and V_j is the output of neuron j. In the simplest, non-graded formulation, the output of neuron i is $V_i = V_i^1$ if $x_i > U_i$ and V_i^0 otherwise; where U_i is the threshold for the neuron i. An energy function such as

$$E = -\frac{1}{2} \sum \sum_{i \neq j} T_{ij} V_i V_j - \sum_i I_i V_i + \sum_i U_i V_i \tag{2.4}$$

may be associated with the network [Hopfield, 82]. Then the change in the energy, ΔE, due to the change in the output of neuron i is

$$\begin{aligned} \Delta E &= -[\sum_i T_{ij} V_j + I_i - U_i] \Delta V_i \\ &= -[x_i - U_i] \Delta V_i \end{aligned} \tag{2.5}$$

The above quantity is always negative because if $x_i > U_i$, then ΔV_i is positive; otherwise both of them are negative. Thus any change in V_i lowers the energy function. Since E is bounded, the system eventually reaches a stable state when no more outputs change. A similar expression for the energy function can also be obtained for neurons with graded response [Hopfield, 84].

2.3.2 Learning

The information content in a neural network resides in the connection strength. Learning is the process of adjusting the connection strengths or the weights in such a way as to produce a set of desired outputs. Learning can be broadly classified into supervised and unsupervised learning. In supervised learning, inputs

are presented along with a set of teaching inputs. Weights are adjusted step by step under the supervision of the teaching inputs so that the network will produce a correct output pattern whenever the trained input pattern is presented. In unsupervised learning, there is no teaching input. However, the network learns by capturing the regularities of the input patterns and responding to any special feature that may be present in the input patterns. A brief review of the back propagation learning scheme (supervised learning) and competitive learning (unsupervised learning) follows next. A detailed discussion on these two learning schemes can be found elsewhere [Rumelhart et. al., 84].

Backpropagation neural networks are strictly hierarchical feedforward multilayered networks (figure 2.3). The first layer is the input layer which receives external inputs and feeds the outputs to the next layer of hidden units. Any layer can receive inputs from the layer just before it and can project the outputs to the layer immediately after it. There may be more than one hidden layer and one output layer. Hidden and output units, employing semilinear activation rules, are useful for capturing higher order regularities. Besides these units, there may also be bias units which are always on and are connected to the hidden and output units.

Backpropagation learning involves two phases of computation. It basically minimizes (gradient descent) the sum squared error over all the output units and all the training patterns. Inputs are presented and the network computes the outputs (O_{pj}). These outputs are then compared to the desired or the teaching inputs (t_{pj}) to generate the error signal δ_{pj} where the suffix p represents any pattern p and j is any unit. Weights are then adjusted for all the connections feeding the output layer according to

$$\Delta_p W_{ij} = \eta \delta_{pj} O_{pi} \tag{2.6}$$

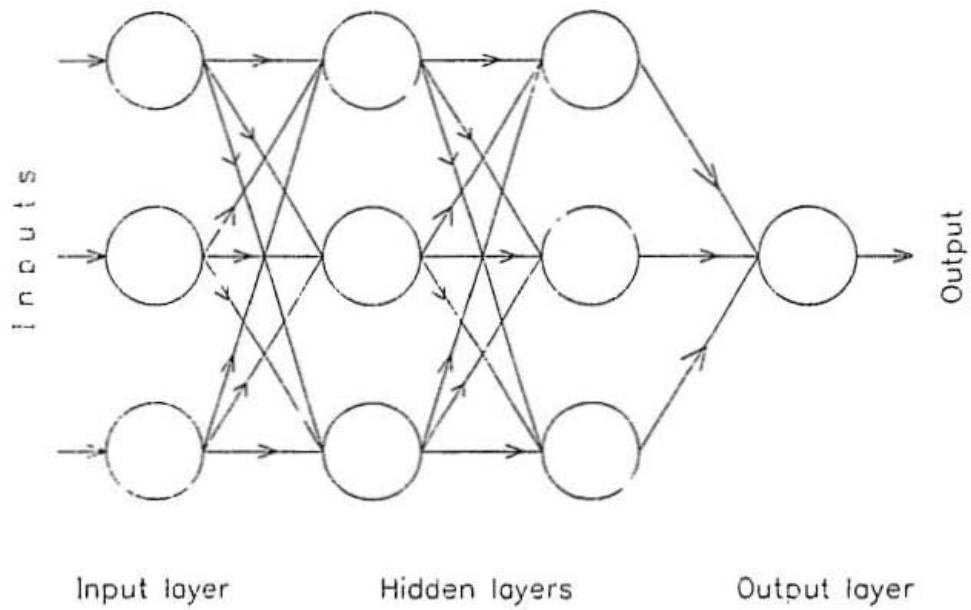

Figure 2.3: A typical multi-layered feedforward network.

where η is the learning rate and O_{pi} is the input to the unit j from the unit i for the pattern p. It can be shown that for the output units

$$\delta_{pj} = (t_{pj} - O_{pj}) f'_j(act_j) \tag{2.7}$$

where f' is the derivative of the activation function. δ's are then computed for the penultimate layer according to

$$\delta_{pk} = f'_k(act_k) \sum_m \delta_{pm} W_{mk} \tag{2.8}$$

where m's are the units connected to the unit k. Thus the error is propagated back one layer. By utilizing the recursive formula of equation (2.8), the error can be computed for any units in a layer and the weights are adjusted according to the equation (2.6). It is important to note that the patterns are required to be presented repeatedly in order to generate the proper internal representation.

The typical activation function employed by the hidden and the output units is given by

$$O_i = \frac{1}{1 + e^{-act_i}} \tag{2.9}$$

This is a sigmoid function which is differentiable as well. The derivative of f, f' is given by

$$\frac{\partial O_i}{\partial act_i} = O_i(1 - O_i) \tag{2.10}$$

This derivative is maximum for $O_i = 0.5$ and since the change in weight depends on this derivative, weight change will be maximum for the units with outputs near the mid-range.

In the competitive learning method, units are also organised in a hierarchical layered fashion. Any units can receive inputs from all the units in the layer immediately below and can feed the output to all the units in the next upper layer, through excitatory connections only. All the units in a layer are grouped

together into a number of clusters. Units in a cluster inhibit each other so that only one unit in a cluster is active ("winner-take-all" strategy). Each unit has a fixed amount of weight distributed over all the input lines i.e. $\sum_j W_{ij} = 1$. Learning is achieved by shifting weights from the inactive input lines to the active ones. If a unit wins, then each of the input lines gives up a proportion (K) of its weight which is redistributed among the active lines. That is

$$\begin{aligned} \Delta W_{ij} &= 0 \quad \text{if unit i loses} \\ &= K\frac{L_j}{n} - KW_{ij} \quad \text{if unit i wins} \end{aligned} \tag{2.11}$$

where $L_j = 1$ if the input line from unit j is active and n is the total number of the active units. However, if the input patterns have few active components, then some of the lines may never be on and the corresponding unit may never win. In order to remove that constraint, the weight can also be changed according to the above equation even if the unit loses, but at a much lower proportion. This at least, will enable the unit to be in the competition. It can also be achieved by changing the threshold in such a way that the unit becomes more sensitive when it loses and becomes less sensitive otherwise.

2.4 Concluding remarks

In this chapter, biological neurons and neural networks have been discussed very briefly. Simple models of the artificial neural networks have been presented along with two learning schemes to introduce basic ideas about neural networks and the relevant terms that would be used throughout the rest of the thesis. With this done, the next chapter is devoted to a literature review.

Chapter 3

Literature Review

3.1 Introduction

Research concerning neural networks can be traced back as far as the 1940s. Since then (except for a brief period in the end of the 60's and the beginning of the 70's) a lot of work has been done on neural networks but mostly involving theoretical studies and computer simulation. Simulation of large neural network is very slow - mostly because of large connectivities among the connecting elements and sequential calculation and updating of neural states. The actual promise of neural networks, however, is in specialized hardware, especially in microelectronic circuits. Then one can possibly exploit the speed and power of neural network and go for practical applications. The major obstacles in realizing neural networks in silicon were the lack of available technology to do so and sufficient knowledge on structures and behavior of neurons in nervous system. However, a great deal of work has already been done on the nervous system and the advent and rapid progress of very large scale integration (vlsi) systems has made it possible these days to realize large neural networks in silicon. A number of researchers are working on the design and implementation of neural networks and a large number of papers has already been published.

Reviewing of this literature can be done in a number of ways. One way is to tackle each of the design issues separately and do a comparative study on different approaches. Alternatively, the designs can be grouped together on the basis of the technologies (e.g. digital, analog, mixed analog-digital etc.) and a study done of each group. Since the proposed circuits are analog, stress will mostly be on different implementation issues of analog circuits including mixed or analog-digital approach. Some of the major problems of pure digital design and some clever solutions of these problems will also be presented.

3.2 Review

3.2.1 Analog implementation

In an analog circuit, the sum of the weighted product can be implemented in a very compact area. This particular aspect has attracted many designers to go to analog circuits. Nevertheless, analog circuits suffer from various problems. First of all noise immunity and immunity to process variability is very poor. The other notable drawback is its comparatively low precision. The latter one is particularly problematic for various learning schemes which need weight adjustment in very small steps. Multiplication is often achieved by the resistors, which suffer from several drawbacks. Current summation is usually employed which can suffer from saturation problem. On top of that, some of the analog circuits tend to be bulky. Analog neural networks have been quite throughly discussed in [Graf et. al., 89].

The major design issues one should consider for the implementation of analog neural networks are :

1) fixed vs. programmable connectivity

2) realization of coupling strengths

3) volatility of connection strengths

4) type of connections

5) size of the neural components

6) ease of fabrication.

7) learning

Fixed vs. programmable connectivity : The computing power of a neural network depends on its connectivity which in turn depends on the problem the network is meant to solve. That is why most of the implementations are application specific. The circuits designed by Graf et. al. [Graf et. al., 87, 88] have programmable connection patterns. The neural output, instead of feeding some other neuron directly, controls two switches. The connection is completed through two other switches which are controlled by the content of two ram cells. The content of the ram determines the type of connection - it can be made excitatory, inhibitory or left unconnected corresponding to a content of +1, -1 or 0. Thus the connectivity of the network can be changed by changing the content of the rams and hence, different configurations can be mapped into the same network.

Coupling strengths : Realization of the coupling strength is an important issue because it determines the network's ability to learn. Most of the earlier designs [Graf et. al., 87, 88], [El-Leithy et. al., 87] used fixed value resistors as the coupling elements. Although this is the simplest way to realize networks, there are a few disadvantages to the approach. First, different connection strengths need different values of resistors and hence, different silicon areas. This prevents the network from having a regular structure as will normally be achieved with fixed size coupling elements (synapses). Then, once fabricated, the resistors cannot be altered any more, freezing the state of the system so that learning cannot take place nor can the system be reprogrammed. Since the patterns to be stored are

often not known a priori, the fixed value resistor approach does not cover as wide a range of applications as one would normally expect from a neural network. Not only that, resistors are expensive in terms of silicon area, particularly the high value resistors required to keep the overall power consumption of the circuit low.

Graf et. al. [Graf et. al., 87] have developed a process by which amorphous silicon can be deposited (as resistive elements) on an otherwise finished chip. Vlsi compatible high value resistors using thin film have also been reported [Hubbard et. al., 86]. These resistors, packed in a chip, can be used to replace the resistor matrix in a network; however, the size of the resistor pack is severly limited by the pin count of the chip. If the precise value of the resistor is not important, diode connected transistors can be used [El-Leithy et. al., 87].

Variable coupling strength has been achieved in [El-Leithy et. al., 87] by adjusting the threshold voltage V_t of the input transistor. V_t depends on a number of parameters, most of which are process dependent (e.g. gate material, gate insulation material and it's thickness, channel doping etc.). It also depends on the bulk (substrate) to source potential V_{SB} of the transistor in a non-linear fashion. By changing V_{SB}, V_t and hence the coupling strength can be changed. However, this requires a variable dc bias for each of the connections and is difficult to realize even for a modest number of neurons. It can be generated on chip, but automatic control will require a rather complex controlling scheme.

A circuit has been implemented using MNOS/CCD principles [Sage et. al., 86] achieving the variable coupling strength in a very elegant way. The circuit works on two concepts - charge coupled device controls the movement of the charge transmitted by a synapse and the MNOS device stores the synaptic weighting value. The charge packet released by the synapse is modulated by the trapped charge under the MNOS gate and a metered quantity is available at the neural output

for generating the activation. By the application of the external voltage, variable amounts of charge can be stored in the nitride layer of the MNOS structure thus achieving different coupling strengths or weights.

Variable coupling strength can also be achieved ([Murray et. al., 89]), by dynamically storing the charge on a capacitor representing the weight voltage. [Brownlow et. al., 90] have used switched capacitor techniques to realize fully programmable weights. Weights are stored in capacitors and are switched by transistors with speeds determined by the incoming pulse rates. Bipolar weights have been realized in [Schwartz et. al., 89] by storing the weights differentially on a pair of capacitors. This scheme also considers weight decay and has achieved 10 bits of analog depth for the weights.

Programmable bistable switches/resistors based on different crystalline materials of Bismuth oxide have been reported [Spencer, 86]. By applying pulses, the conductivity of the material can be increased by several orders of magnitude. It can be brought back to the initial insulating state by applying negative pulses. When electric field is applied, vacant oxygen sites are created which contribute to the conductivity. By suitable biases and applying pulses, resistivity of the required value can be obtained and hence can be used as a programmable connection elements for the neural networks. This scheme seems to be an interesting proposition but requires a lot of improvement on the metallurgy of these materials so that it would be possible to realize a large scale array with identical switching characteristics.

A two quadrant multiplier with a digital weight scheme has been described in [Hollis et. al., 90]. Weight is represented by a set of parallel binary weighted (W/L ratio varies in binary fashion) current sources.

Floating gate technology seems to be the most viable weight storage scheme.

It has been successfully used in the ETANN chip [Video, 91] and gives 6-8 bits precision. Under this scheme [Sze, 81], charge is injected from the silicon across the first of the two insulators and stored in the floating gate giving rise to a threshold voltage shift. Programmability is easily achieved by storing different amount of charge in the floating gate. MNOS is a similar device but has a different structure.

Volatility of the connection strength : Nonvolatility of the connection strength is important because once a proper set of weights is learned, it should be retained for future use. Resistors are best suited for this purpose. This is also easily achieved in Sage's approach [Sage et. al., 86]. The charge that is trapped under the nitride layer has a very high retentivity at the normal operating conditions. The floating gate approach or FAMOS is also very much suitable for long term charge storage. In [Murray et. al., 89], the charge storage being dynamic, there is steady leakage of charge from the capacitor. Periodic restoration of charge is done [Brownlow et. al., 90] from off chip ram through a digital to analog converter.

Type of connections : Most of the papers being discussed here use both excitatory and inhibitory synapses. One common way of realizing inhibitory synapses ([Graf et. al., 86, 88], [Tank et. al., 86, 87]) is to use the inverted output of the neuron. In the paper [Verleysen et. al., 89], a simple digital control drives all excitatory current through one line and all inhibitory current through the other line depending on the sign of a control line. Inhibition in [El-Leithy et. al., 87] is achieved by using PMOS transistors. Inhibition is also achieved in [Murray et. al., 89] by removing charge from the capacitance, the voltage across which represents the activity of the neuron. However, only one type, namely the excitatory connection has been achieved in [Sage et. al., 86].

Size of the neural components : The area of the neural components has

to be small in order to accommodate a large useful network in a chip. Since the number of synapses is usually much larger than that of neurons, one has to minimize the size of the synapse. The MNOS/CCD circuit is very compact and so are circuits described in [Brownlow et. al., 90]. In [Cotter et. al., 88], few neural building blocks have been designed which can be used advantageously to realize neural networks in vlsi. Analog computers with a number of vlsi chips in conjunction with a host computer have been discussed in [Eberhardt et. al., 89], [Mueller et. al., 89]. A number of chips can be connected together to realize a large network. Functionally both the schemes are quite competent but they require complex control and timing and can accommodate only a small number of neural components per chip.

Ease of fabrication : One has to be careful about choosing the basic components so that they can be fabricated using the widely available fabrication processes. The circuit in [Sage et. al., 86] employs special fabrication techniques for realizing the MNOS device. Resistors are realized in [Graf et. al., 86] by a special fabrication technique and also in [Hubbard et. al., 86] although it was claimed to be a vlsi compatible process. Bistable switches and resistors [Spencer, 86] also require special fabrication procedures.

Learning : Since the work presented in this thesis does not consider learning, learning capability of different circuits will not be discussed.

Pulsed analog neural circuits fall under the analog category and are one of the most attractive candidates for neural networks. A variety of techniques such as pulse width modulation, pulse height modulation, simple gating etc. can be used to multiply the pulse stream by the weight voltage. Pulsed circuits have been reviewed quite nicely in [Murray et. al., 91] and pulse height modulation seems to be the best candidate for this purpose. Under this scheme, analog weight voltage

is stored on a capacitor and incoming pulse is modulated by this weight voltage through a MOS transistor [Murray et. al., 89].

3.2.2 Digital implementation

A pure digital approach to the implementation of neural networks suffers from a few drawbacks even though it has quite a few positive points that makes it an attractive candidate for vlsi system. Registers are needed for storing the weights. Digital multipliers and adders are required to obtain the sum of the weighted product. All these are expensive in terms of silicon area. Time sharing is one way of taking care of this problem but this calls for a complex control scheme and at the same time reduces computational speed. On the other hand digital circuits are robust with respect to noise and process variations. They are well suited for applications where precision is more important than the complexity or the size and are particularly very well suited for various learning schemes.

The digital approach is problematic for a fully connected network and is more suited for a layered network. This is so because at each connection, one needs an adder and a multiplier and they are expensive. However, different approaches can be taken to overcome these problems. The next few paragraphs deal with some of the innovative approaches for realizing the sum of the weighted products.

A digital neuro-chip with six neurons and eighty four synapses has been described in [Hirai et. al., 89]. The neurons operate asynchronously and several chips can be connected together to realize networks of any arbitrary size. Synaptic weights are programmable (64 levels) and can be set or monitored by a host computer. The incoming pulse density is transformed to a density proportional to the weight by the rate multiplier. An up - down counter is used to realize excitation and inhibition and a rate multiplier is used to generate the output pulses.

This scheme, even though unique in its conception, is very bulky.

A digital pulse density modulation circuit has been designed and described in [Tomberg et. al., 90]. Each chip can be used as a stand-alone device or can be cascaded to form a larger network. Instead of using normal binary arithmetic numbers, pulse density arithmetic (where each bit has exactly equal weight) numbers have been used resulting in a simple control and arithmetic. This has been achieved at the expense of a greater number of bits than is required in normal binary arithmetic. Multiplication is achieved by doing x-or and addition is bit serial thus the computation time is directly proportional to the number of neurons (for a fully connected network).

The circuit given in [Blayo et. al., 89] has realized a fully connected network with systolic architecture. For N neurons, 2N steps are required to compute the sum of the weighted product. The performance can be improved by introducing pipelining but the approach needs very complex circuitry and controls.

A multilayered neural architecture using cellular arrays has been given in [Faure et. al., 89]. Each array is connected to it's four adjacent neighbors through eight bi-directional buffers. Each cell consists of a routing part and a processing part and by loading appropriate messages, any cell can be logically connected to any other cell.

In another bit serial approach [Butler et. al., 89], each synaptic element adds its share of weighted product to the partial sum line running down the synaptic column. The output state is restricted to 5 different levels and the multiplication by the weight is achieved by shifting the binary weight.

A different approach has been taken in [Weinfield, 89] where the neural output states are stored in a circular shift register which can be simultaneously accessed by all the neurons. A simultaneous partial potential is thus obtained at each shift

operation of the register (for a fully connected network). But the whole neural circuit as such is very bulky, it includes an adder, comparator, sixty-four 9 bit weight storage areas etc.

3.3 Concluding remarks

This chapter dealt with different circuits and implementation techniques for the neural networks. Some circuits have certain advantages in some of the design aspects but disadvantages in others. Floating gate technology seems to be the most suitable candidate for programmable, long term analog weight storage. The synaptic circuit has to be compact compared to the neuron in order to achieve high integrability. With these in mind, the next chapter deals with the design philosophy and the motivation behind the design of this particular kind of circuitry.

Chapter 4

Design Philosophy & Proposed Architecture

4.1 Introduction

In any neural network, whether artificial or biological, the number of synapses is much higher than the number of neurons. In a fully connected network of n neurons, the number of synapses grows as n^2. When the implementation in silicon is at hand, one has to consider the implication of scale very carefully. The number of synapses per neuron should not be limited by any circuit constraints. The pulsed analog neural circuits being proposed here, have distinct advantages over most of the existing neural circuits.

4.2 Objectives

The main objectives of this design approach are

- to minimize the synaptic area
- to develop an efficient way of adding a large number of synaptic outputs together

- to design standard cells which can be put together to realize any neural circuits independent of synapse to neuron ratio – i.e. scalability.

4.2.1 Motivations

Minimization of the synaptic area is important because the synapses predominate in any neural circuit. However, it is problematic to add the outputs of a large number of synapses together and then feed the sum to the neuron input. Conventional digital and analog circuits suffer from various drawbacks and they have already been discussed in chapter 3.

Pulsed analog circuits seem to be the most effective way of realizing very compact and efficient synapses [Murray et. al. 91]. Under this scheme, the neural state is represented by a train of digital pulses the frequency of which depends on the input activation. The height of the incoming pulses is modulated by a locally stored analog weight voltage. If the width of the pulse is narrow, the modulated current can be thought of as a charge packet and can be dumped on to a capacitor. If a large number of synapses are connected together, more and more charge will be dumped on the capacitor thereby increasing the membrane voltage steadily. In order to overcome the saturation, the capacitance has to be increased in proportion to the number of inputs.

Another advantage of this approach is that the information content is in the frequency of the neural output pulse, not in its height. So the output can be routed to a distant synapse very easily. Not only that, the pulses, being essentially digital, can be restored by means of digital buffers while being routed over a large distance. The same buffer can also be used to handle the fanout problem.

The other point worth mentioning here is that the synaptic circuits are either excitatory or inhibitory but not both. That is, they cannot move back and forth

between excitation and inhibition as they can in most of the existing neural network models. So far, this kind of synapse has never been observed in biological neurons [Personnaz et. al., 86]. This bipolarity can easily be handled in digital circuits but is problematic in analog implementations. For instance, where a multiplier is used for the weight circuit, one needs a four quadrant version instead of a single quadrant if bipolarity is allowed.

Finally the transformation of charge packets in the synapse is achieved by applying the pulses to a MOS transistor whose gate is held at the weight voltage. So the amount of the charge being dumped on the capacitor depends on the rate (and width) of incoming pulses and the weight voltage. However, since the MOS transistor is inherently nonlinear, scaling the weight voltage will not scale the neural output linearly. So far, there is no evidence that linearity is maintained in the biological system. Moreover, all the learning mechanisms employ some sort of feedback where the weight is changed till the correct output is obtained. This does not demand linearity so long as monotonicity is preserved. Even if it turns out that linearity is the rule in biology, it may be worthwhile to allow non-linearity in order to achieve a very compact and efficient synapse.

4.3 Proposed Architecture

Figure 4.1 shows the basic architecture of the proposed pulsed analog neural network. For the reason described later (chapter 7), two different types of neurons have been designed. The first one is the standard neuron used in the hidden layer and the output stage. It consists of a comparator and a pair of pulse generators which emit one pulse each, every time the input activation goes past the threshold. The other type is for input neurons which are rate generators. One kind fires at a maximum rate with an input voltage of 5 volts and gradually decreases the

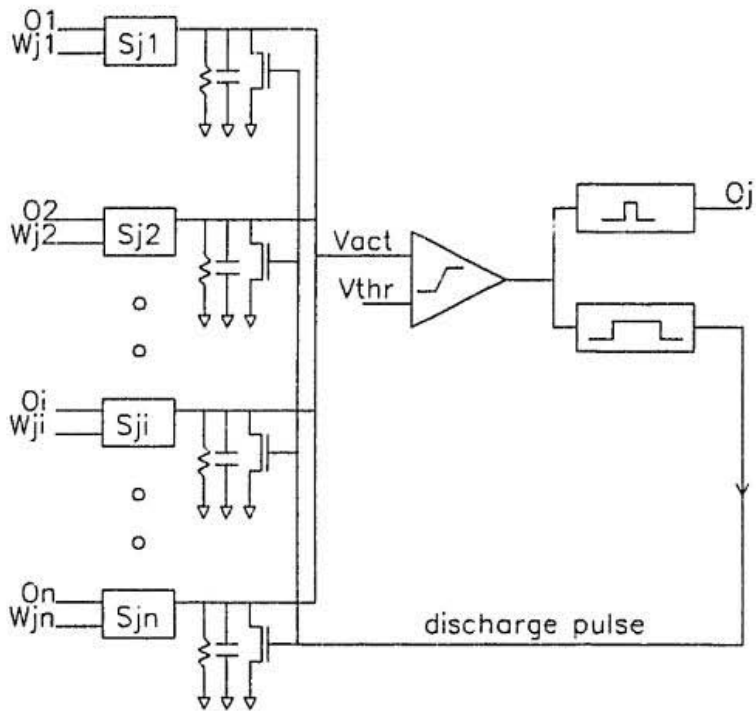

Figure 4.1: Block diagram of the proposed auto scaling pulsed neural network.

rate as the input voltage goes down. Whereas, the input neuron with a circle at the input (not shown, please refer to figure 7.1) is the inverting type which fires at the maximum rate when the input is zero and decreases the rate as the input voltage goes up. This second type of neurons exists in the retina which respond to darkness instead of the light and are called dark cells.

The channel resistance of the synaptic transistor is controlled by the weight voltage, thus controlling the amount of charge flow to and from the membrane capacitance. If the synapse is excitatory, charge is added to the capacitor while it is removed for the inhibitory synapse. To achieve scalability, the membrane capacitance has been distributed over the synapses. So, the total capacitance of the neuron depends on the number of synapses attached to the neuron. If the capacitance were included in the neuron instead, the size of the capacitance would have to be changed depending on the number of synapses in order to avoid the problem of saturation. Another advantage is that each synapse adds or subtracts its own share of charge thus avoiding the problem of current density build-up due to simultaneous arrival of pulses. Since all these capacitors are in parallel, the addition essentially turns into a single minimum size line or bus representing the input activation or the membrane voltage.

The other point to be noted here is that the inhibitory synapses do not contain any capacitors and the scaling is applied only to the excitatory synapses. Since the inhibitory synapse removes charge from the total membrane capacitance thereby making it harder for the neuron to overcome the threshold, there can be no neurons with inhibitory synapses only. They exist only to inhibit the excitation and this is achieved by removing the charge, not by increasing the capacitance. However, the scaling of the inhibitory synapses is obtained in the secondary level in the sense that there must be more inhibitory synapses as the number of excitatory

synapses goes up.

Once the neuron fires, the associated synaptic capacitors are discharged so that the charge integration cycle can start once again. Scalability has once again been achieved by distributing the discharge transistors in the synapses. Though a large number of transistors are required, all of them operate in parallel as a single wide transistor whose width is scaled up by the number of synapses. Whereas if a single transistor is used in the neuron, its width has to be adjusted according to the number of synapses - thus making it impossible to go for the standard cell approach. An extra set of connections from the neuron output to the synapses are needed in order to broadcast the discharge pulse, but they can run in parallel to the wires carrying the synaptic outputs together to the neuron. Thus, the channel will be slightly wider to accomodate a two wire bus instead of one.

Since the amount of charge being dumped on the capacitor depends on the pulse width, it is required that the output pulse be narrow. However, for proper discharge operation, the discharge pulse has to be significantly wider. That is why two pulse generators have been included.

Since there will be n neurons and s synapses where s$\gg$n, synapse S_{ij} will receive a discharge pulse from the neuron i and an output pulse from the neuron j. Thus, on an average, output and discharge pulses are to be fed to almost equal number of synapses. Since this number can be very large, fanout problems have to be handled. A digital buffer has been designed having the same height as the synapse. Two buffers occupy roughly the same area as a synapse. Consequently, the buffers can be inserted in the synaptic ranks very easily, and the signals can be routed through the buffers.

The resistor in the synapse represents the leakage for the proper operation of the neuron. Without it, the neuron integrates the incoming pulses indefinitely

reducing the firing rate along a layered network. By distributing the leakage along with the capacitor, the time constant has been made independent of the scale.

Although some circuits for weight manipulation have been designed, not much work has been done on that and consequently it will not be included in the thesis. The floating gate technology seems to be the most promising candidate for the implementation of the efficient weight storage (chapter 3) and the synapse has been designed with that in mind.

4.4 Concluding remarks

In this chapter, design philosophy for the auto scaling neural architecture has been presented. There has been significant deviation in the proposed architecture from the existing ones. Membrane capacitance has been distributed in the synapses for the purpose of scalability. The neuron fires only one pulse when the input activation exceeds the threshold voltage. A discharge pulse is also generated to discharge all the associated synapses so that the charge integration cycle can start once again. Scalability has once again been achieved by distributing the discharge transistors over the synapses. Thus scability has been achieved at the expense of a slight increase in the synaptic area.Two different input neurons have also been proposed for interfacing networks to the external inputs. The neural architecture having been proposed, the next chapter deals with the design and analysis of the individual blocks.

Chapter 5

Circuit Design and Analysis

5.1 Introduction

This chapter contains the designs for different neural circuits, namely the excitatory synapse, the inhibitory synapse and the standard and the input neurons. Simulation of the circuits using Spice, and a mathematical analysis of each of them has also been provided.

5.2 Excitatory Synapse

5.2.1 Circuit Description

The excitatory synaptic circuit is shown in figure 5.1. The circuit uses two minimum size NMOS transistors in series. The excitation voltage V_{ex} is applied to the gate of the first transistor M1 and the second transistor M2 is gated by the weight voltage V_{wt}. The drain of M1 is pulled high. The output of the synapse is the membrane voltage V_m taken across the membrane capacitance C_m. Transistor M3 (minimum size again) is the discharge transistor which discharges C_m whenever the neural input activation (or the membrane voltage) exceeds the threshold voltage. Transistor M4 is a long transistor which generates the leakage required for the proper operation of the synapse.

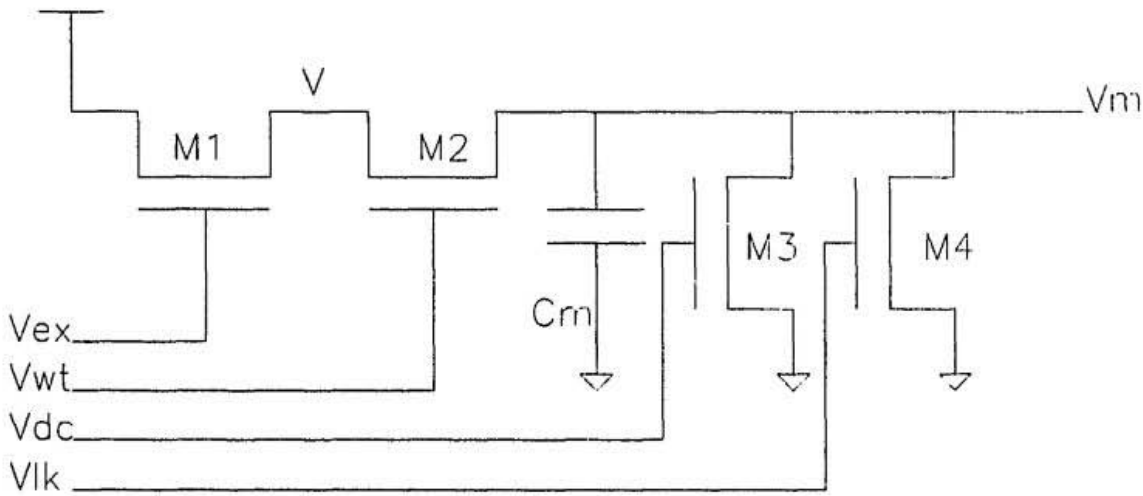

Figure 5.1: Schematic of the excitatory synapse. All transistors have W=5.4μ and L=3μ except for M4 which has L=20.6μ.

If only one transistor (M1) were used instead of M1 and M2, the excitatory signal would have to be applied to the drain of M1. Since a neuron may be connected to many synapses, this would require a large driving capability from the neuron output. Not only that, when the input signal V_{ex} is off and V_m is greater than zero, the drain becomes the source. The gate being pulled to the weight voltage, there will be a steady current flowing from the capacitor C_m to the ground. Since the gate current is negligible, the above approach takes care of both problems.

The amount of current flowing through M1 and M2 depends on the gate voltage of M2 and the voltage across C_m. By controlling the gate voltage V_{wt}, the amount of charge that would be dumped on the capacitor can be controlled. So the effect of the excitatory pulse from neuron u_j to neuron u_i through the synapse S_{ij} depends on V_{wt}. V_{wt}, therefore, is the strength of connection between the two neurons.

Transistor M1 is always in saturation because the drain is being pulled high. For most of the useful weight voltage range (described later), M2 will be in the linear region. However, the charging current is somewhat less due to higher threshold voltage because of non-zero bulk to source potential. This is not a problem for the proper operation of the proposed circuit but can be taken care of by tying the substrate to the source potential of the transistors M1 and M2. Charging current can also be increased by increasing the width of M1 and M2. But minimum sized transistors are good enough for this application. Minimum size also reduces the parasitic capacitances.

Charge is dumped only during the time V_{ex} is high. When V_{ex} is low, M1 is cutoff but a small leakage current flows through the reversed biased diodes between the source, the drain and the substrate. This leakage current is very small ($\approx$ 10pA) and can be ignored. This is because synapses will operate in

parallel and the amount of incoming charge ($\approx 5\mu$A/excitatory synapse) will be much more than the charge lost due to unwanted leakage.

5.2.2 Circuit Design

If the leakage is not included in the synapse, the neuron output period is

$$T = T_d + T_c \tag{5.1}$$

where T_d and T_c are discharge and charge time respectively. If there are n excitatory synapses, and if I_{av} is the time averaged charge arrival rate, then the total arrival rate is nI_{av}. For n synapses, the total membrane capacitance is nC_m. So the average charge time is

$$T_c = nC_m \frac{V_t}{nI_{av}} = C_m \frac{V_t}{I_{av}} \tag{5.2}$$

where V_t is the threshold voltage of the neuron. The average firing rate is

$$R = \frac{1}{T} = \frac{1}{T_d + C_m \frac{V_t}{I_{av}}} \tag{5.3}$$

Normalizing the average firing rate to the maximum firing rate ($1/T_d$) and the average rate of charge arrival to the average maximum arrival rate I_{max}. one gets

$$\frac{R}{1/T_d} = \frac{1}{1 + C_m \frac{V_t}{\frac{I_{av}}{I_{max}} Q_{max}}} \tag{5.4}$$

I_{max} is the theoretical maximum rate and is given by

$$I_{max} = \frac{Q_{max}}{T_d} \tag{5.5}$$

where Q_{max} is the maximum rate of charge transfer by a single synapse excited at the rate of $1/T_d$ and with a weight voltage of 5 volts. The above equation is plotted in figure 5.2 and basically represents the activation function. The curve does not exhibit the two decision states normally present in neural networks. The

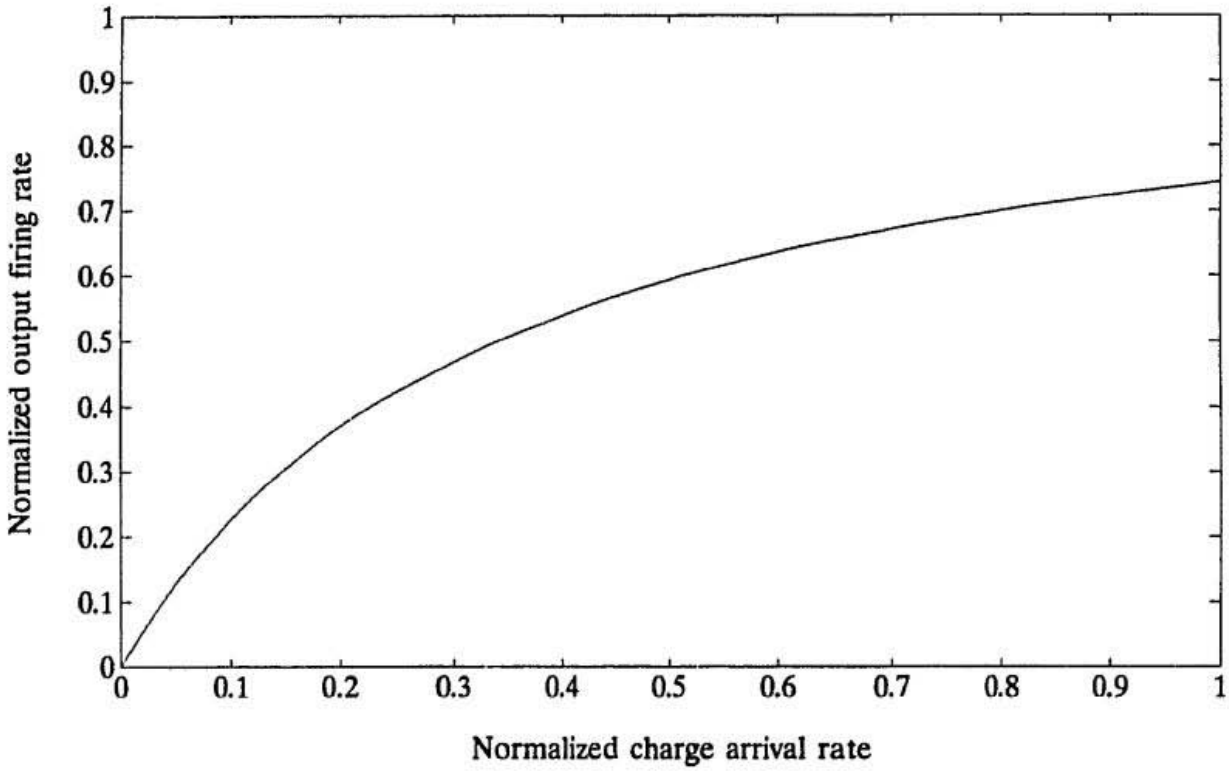

Figure 5.2: Normalized firing rate of the neuron without leakage.

implication is that a minimal charge arrival rate should be maintained (in order to overcome the leakage as in the biological neuron) to generate an output.

However, if a resistance R_{lk} is added in parallel to the capacitance,

$$I_{av} - \frac{V}{R_{lk}} = C_m \frac{dV}{dt} \tag{5.6}$$

The steady state solution is $V = I_{av}R_{lk}$. If $I_{av}R_{lk} < V_t$, the device will never fire. V can be represented by the standard exponential equation

$$V = I_{av}R_{lk}(1 - e^{-\frac{t}{R_{lk}C_m}}) \tag{5.7}$$

The charging time T_c is the time taken to charge up to V_t, the threshold voltage. So, the output firing rate is

$$\begin{aligned} R &= \frac{1}{T_d - R_{lk}C_m ln[1 - \frac{V_t}{I_{av}R_{lk}}]} \quad \text{if } I_{av}R_{lk} > V_t \\ &= 0 \quad \text{otherwise} \end{aligned} \tag{5.8}$$

Normalizing the equation once again, one gets

$$\frac{R}{1/T_d} = \frac{1}{1 - \tau ln[1 - \frac{I_t}{I_{av}/I_{max}}]} \tag{5.9}$$

where $\tau = \frac{R_{lk}C_m}{T_d}$ and $I_t = \frac{V_t/R_{lk}}{I_{max}} = \frac{V_t T_d}{R_{lk}Q_{max}}$. I_t is current threshold which determines the firing instance. Figure 5.3 shows a series of plots for I_t=0.3 and various values of C_m (that is different values of τ). It can be seen that τ influences the curvature of the plots. This activation function clearly shows two decision states but is not sigmoid. This type of activation function has been described in [Rumelhart et. al., 84].

The capacitor C_m, apart from the bulk membrane capacitance, includes the parasitic capacitances as well. These parasitic capacitances stem from the bulk to drain capacitance of transistors M3 and M4 and the body to source capacitance

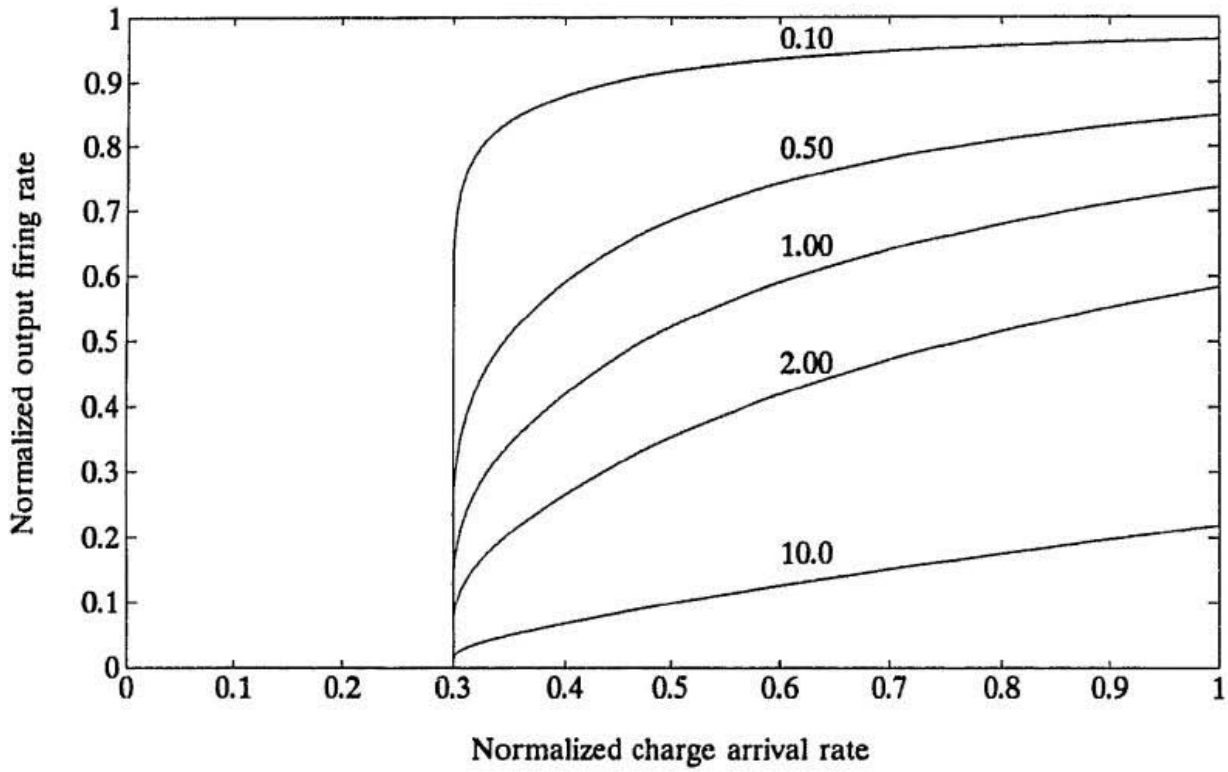

Figure 5.3: Activation curves for different values of τ (when leakage is included).

of M2. These capacitances are voltage and geometry dependent. The expression for these capacitors are given by ([Geiger et. al., 90])

$$C_{BD} = \frac{CJ.A}{[1-(V_F/\phi_B)]^{MJ}} + \frac{CJSW.P}{[1-(V_F/\phi_B)]^{MJSW}} \tag{5.10}$$

The minimum size geometry is $L = 3\mu m$ and $W = 5.4\mu m$. $W = 5.4\mu m$ is chosen to avoid the dog bone effect at the drain and the source contact points. The total parasitic capacitance comes to around 50 to 60 fF. Since this is quite variable (process as well as operating point dependent), membrane capacitance has been chosen to be 100 fF or 0.1 pF, bringing the total membrane capacitance to around 0.15pF. The incoming pulse width has been set to 6.5ns so that a single synapse with a weight voltage of 5 volts can make the neuron fire one pulse but not if the weight voltage is less than 3.6 volts. The threshold voltage of the neuron has been chosen to be equal to 1.5 volts which is half way between the upper and lower values of low and high logic levels. The useful weight voltage is from V_{wt}=2.5 volts to 5 volts. The lower value is due to the fact that M2 conducts (ignoring subthreshold operation) when V_{wt} is more than the neuron threshold voltage (=1.5 volts) plus its own threshold.

The maximum charge that can be delivered to the capacitor by one single pulse has been simulated to be equal to 437 fC. T_d was set to 30ns so that the maximum current that can be provided by a single synapse is 14.6 μA. If R_{lk} is chosen to be 500 $k\Omega$ then τ=2.5 and I_l=0.21. This is in good agreement because I_{max} is the upper bound of the current.

Transistor M4 replaces R_{lk} and has a constant gate voltage of 1.5 volts. M4 will be in saturation for any membrane voltage more than 1.5 - 0.7 (threshold voltage) or 0.8 volt. However, when the membrane voltage is less than 0.8 volt, M4 is in the linear region and the leakage current is less. To compensate for this fluctuation, I_l is set 10% higher than was derived and it's absolute value is

0.23*14.6 μA = 3.4 μA. In saturation

$$I_t = 0.5 * k' * S4 * (1.5 - 0.7)^2 \tag{5.11}$$

where S4 is $\frac{W_4}{L_4}$ and is the shape factor. S4 turns out to be equal to 0.26. With W=5.4 μm, L comes to 20.6 μm.

A Spice simulation of a single synapse with 6.5ns excitatory pulses and 5 volts weight voltage is shown in figure 5.4.

5.2.3 Circuit Analysis

Referring to the figure 5.1, for a weight of 5 volts, transistor M1 will be in saturation and M2 will be in the linear region. The current through M1 is given by

$$I_1 = \frac{K1'}{2}\frac{W1}{L1}(V_{ex} - V - V_{t1})^2 \tag{5.12}$$

and the current through M2 is

$$I_2 = \frac{K2'}{2}\frac{W2}{L2}[2(V_{wt} - V_m - V_{t2}) - (V - V_m)](V - V_m) \tag{5.13}$$

Neglecting the parasitic capacitances at the junction of M1 and M2, one can say that $I_1 = I_2$. The threshold voltage of M1 is given by

$$V_{t1} = V_{T0} + \gamma(\sqrt{\phi + V} - \sqrt{\phi}) \tag{5.14}$$

where V is the source voltage of M1. Similarly, the threshold voltage of M2 is

$$V_{t2} = V_{T0} + \gamma(\sqrt{\phi + V_m} - \sqrt{\phi}) \tag{5.15}$$

Solving the equation for $I_1 = I_2$ and noting that V_t's can be taken to be constant for small time steps, V can be written as

$$V = \frac{1}{2}(a + b) \pm \frac{1}{2}((a + b)^2 - 2[a^2 + 2V_m b - V_m{}^2])^{\frac{1}{2}} \tag{5.16}$$

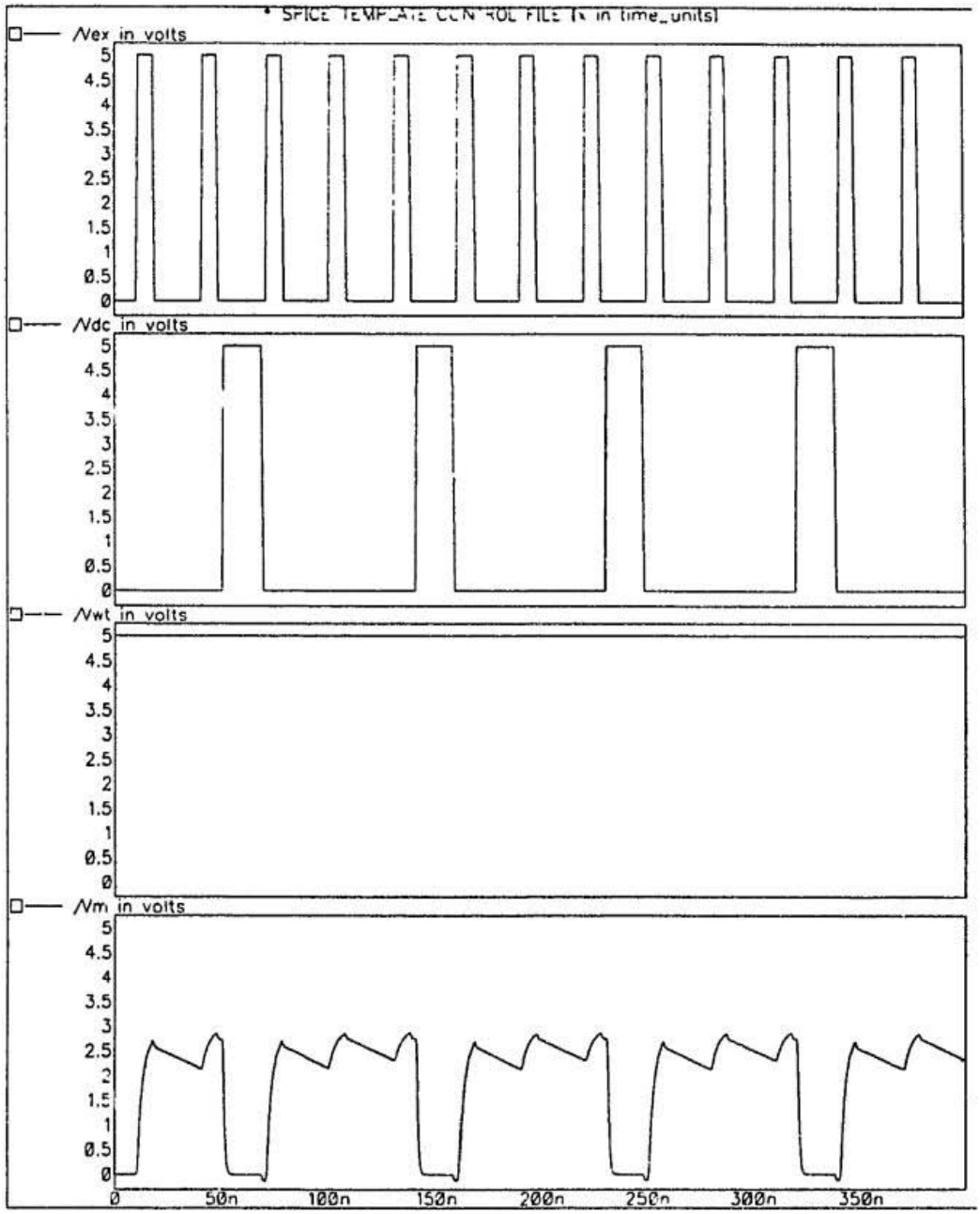

Figure 5.4: Simulation result of a single synapse with a weight voltage of 5 volts and 6.5 ns excitatory and discharge pulses.

where a=V_{ex}-V_{t1} and b=V_{wt}-V_{t2}. This value of V can be put back to the equation 5.13 for I_2 to find the instantaneous current through M2. If the time steps are taken to be small enough, then

$$I_2 = C_m \frac{\Delta V_m}{\Delta t} \tag{5.17}$$

Figure 5.5 shows the output of a small C-program (along with Spice output for a comparison) to compute the output of a single synapse due to the excitatory pulses (width is 7.0ns and period is 30ns). It agrees quite reasonably with the simulation results from Spice. The small deviation is due to the fact that the program does not consider the higher order effects which are present in the Spice level 3 simulation. The other point to be noted here is that the leakage transistor has been omitted.

5.3 Inhibitory Synapse

5.3.1 Circuit Description

The inhibitory synapse is shown in the figure 5.6. It is almost identical to the excitatory synapse without the transistors M3, M4 and the membrane capacitance C_m. The drain of M1 is grounded, so it becomes the source. Since all the synaptic outputs will be tied together to constitute the activation bus for the neuron, application of the pulses at the gate of M1 will result in withdrawal of charge from the total membrane capacitance. This discharge current, however depends on the weight voltage V_{wt} at the gate of M2. Most often a stronger inhibition (compared to the excitation) is required so that the weight voltage will be around 5 volts. This will make M2 operate in the linear region. Since the source of M1 is grounded, M1 will also be in the linear region.

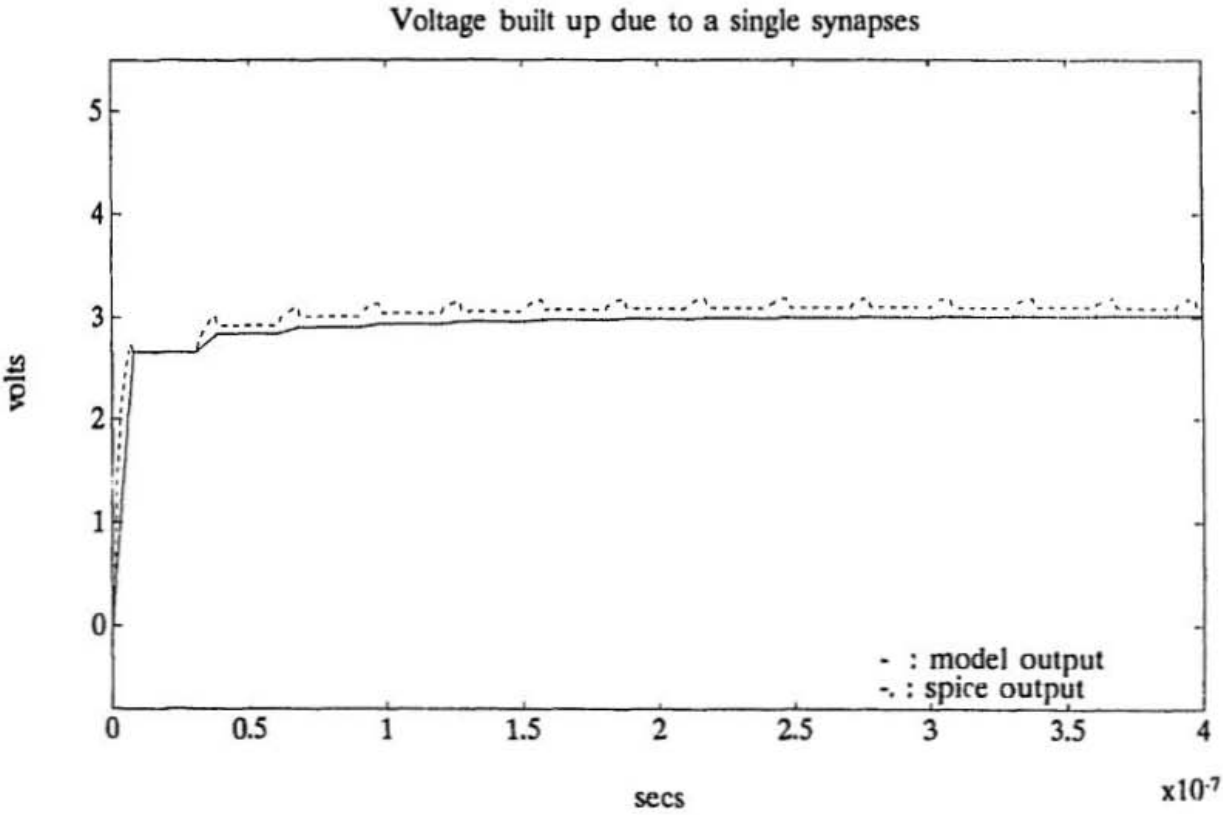

Figure 5.5: Activation voltage due to one synapse using equations 5.13, 5.16 and 5.17. A spice simulation is given for comparision.

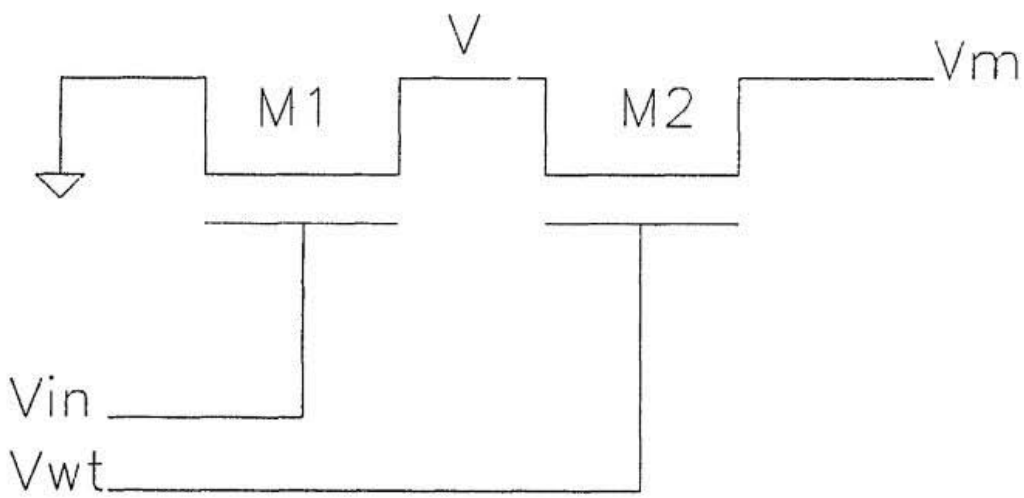

Figure 5.6: Schematic diagram of the inhibitory synapse. L=3μ and W=5.4μ for both the transistors.

5.3.2 Circuit Design

There is no elaborate design procedure for the inhibitory synapse. Both M1 and M2 are minimum sized transistors. A spice simulation of three excitatory synapses and one inhibitory synapse is given in the figure 5.7.

5.3.3 Circuit Analysis

When the inhibitory pulse V_{in} is applied to the inhibitory synapse (figure 5.6) with a weight voltage V_{wt} of 5 volts, an expression for the discharge current can be obtained as follows : for the quantitative analysis, V_{in} will be taken to be the threshold voltage of the neuron or 1.5 volts. If V is the drain voltage of M1, then the current through M1 is

$$I_1 = \frac{K1'}{2}\frac{W1}{L1}[2(V_{in} - V_{t1}) - V]V \tag{5.18}$$

and through M2 is

$$I_2 = \frac{K2'}{2}\frac{W2}{L2}[2(V_{wt} - V - V_{t2}) - (V_M - V)](V_M - V) \tag{5.19}$$

where V_M is the total membrane voltage of the neuron. Equating these two current expressions as in the last section, V can be solved to be

$$\begin{aligned} V &= \frac{1}{2}(V_{in} - V_{t1} + V_{wt} - V_{t2}) \\ &\pm \frac{1}{2}\sqrt{(V_{in} - V_{t1} + V_{wt} - V_{t2})^2 - 4(V_{wt} - V_{t2} - .5V_M)V_M} \end{aligned} \tag{5.20}$$

Here, $V_{t1} = V_{T0}$ and V_{t2} is given by

$$V_{t2} = V_{T0} + \gamma(\sqrt{\phi + V} - \sqrt{\phi}) \tag{5.21}$$

Substituting V in the expression for I_2 will give the value of the discharge current. Finally, it can be noted that if C_M is the total membrane capacitance, then

$$I_2 = C_M\frac{\Delta V_M}{\Delta t} \tag{5.22}$$

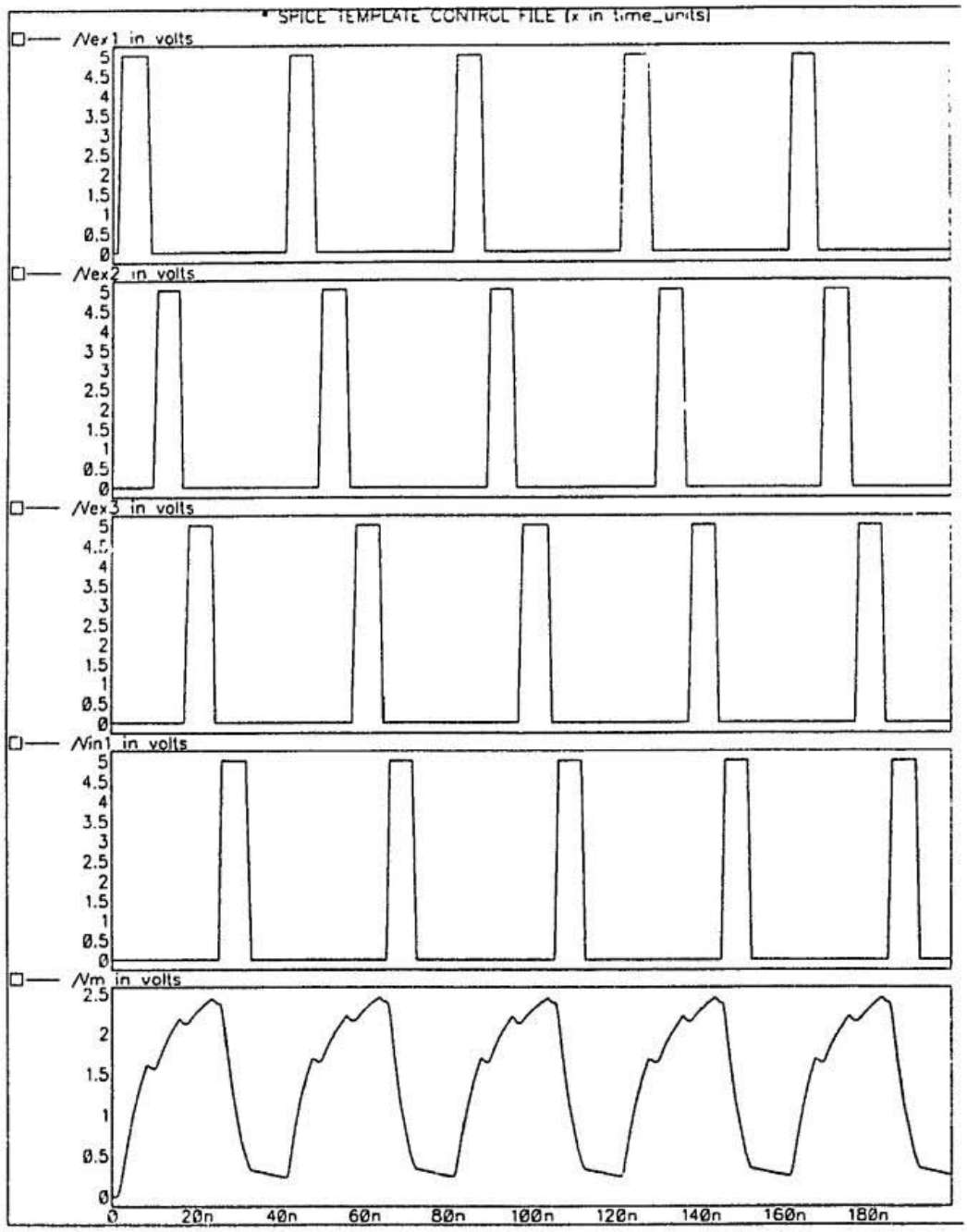

Figure 5.7: Spice simulation of three excitatory and one inhibitory synapses. First four waveforms are input pulses and the last is the output when the synapses are tied together.

From the above equation, one can find the change in the activation voltage due to an inhibitory pulse.

5.4 Standard Neuron

5.4.1 Circuit Description

The standard neuron consists of a comparator and two pulse generators. Whenever the input activation voltage goes past the neuron threshold voltage, the comparator output goes high. This positive transition is detected by the pulse generators which in turn emit one 6.5 ns neural output pulse and one 17 ns wide discharge pulse. This discharge pulse discharges the membrane capacitances and the comparator output eventually goes low. The schematic diagram is shown in figure 5.8.

5.4.2 Circuit Design

The neuron circuit is quite straightforward. A standard n-channel differential amplifier with a p-channel current mirror load is used as the comparator. A second inverting stage is added to increase the gain and the output swing of the amplifier. The output is fed through a standard digital buffer, the output of which feeds the pulse generators. This certainly limits the amplifier load and hence improves the comparator delay. The pulse generators are pulse edge differentiators using the logic gate delays. Small capacitors (0.1 pF) have been added in the inverter chains to increase and achieve the desired delay. Spice simulation of the standard neuron for a ramp input is given in figure 5.9 (a Spice input deck is included in Appendix).

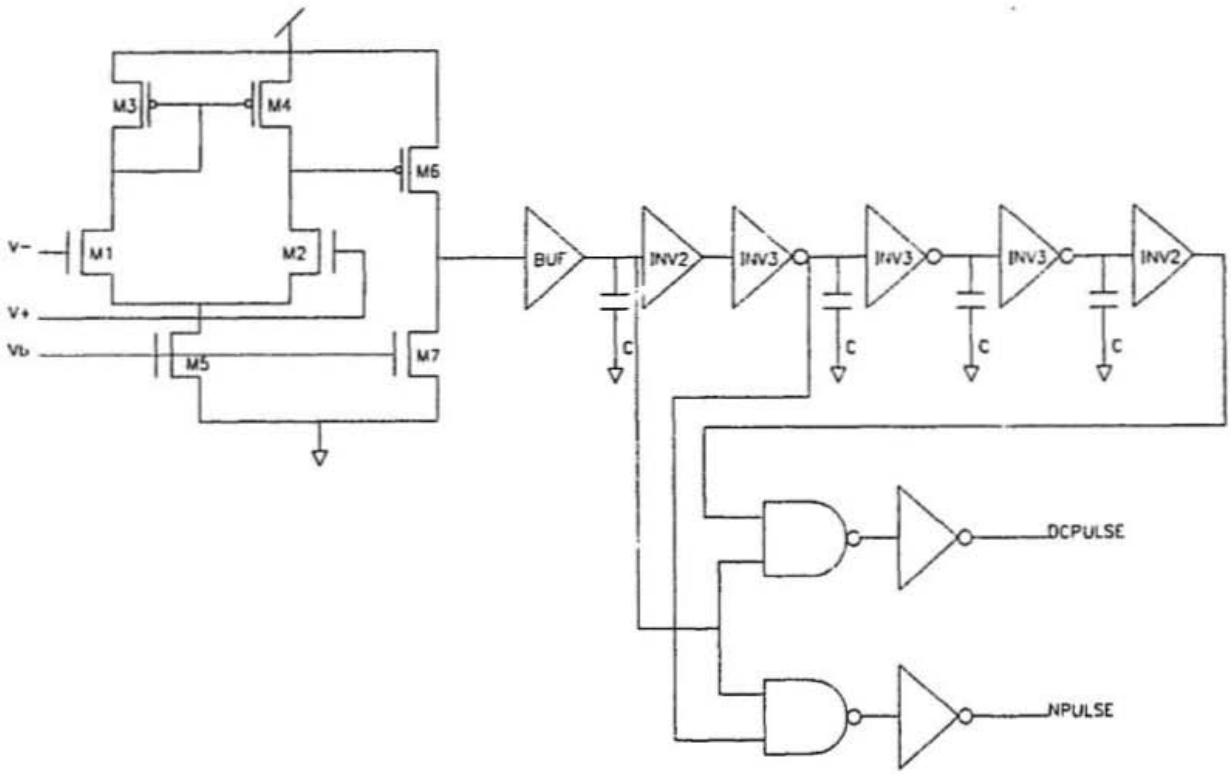

Figure 5.8: Schematic of the standard neuron.

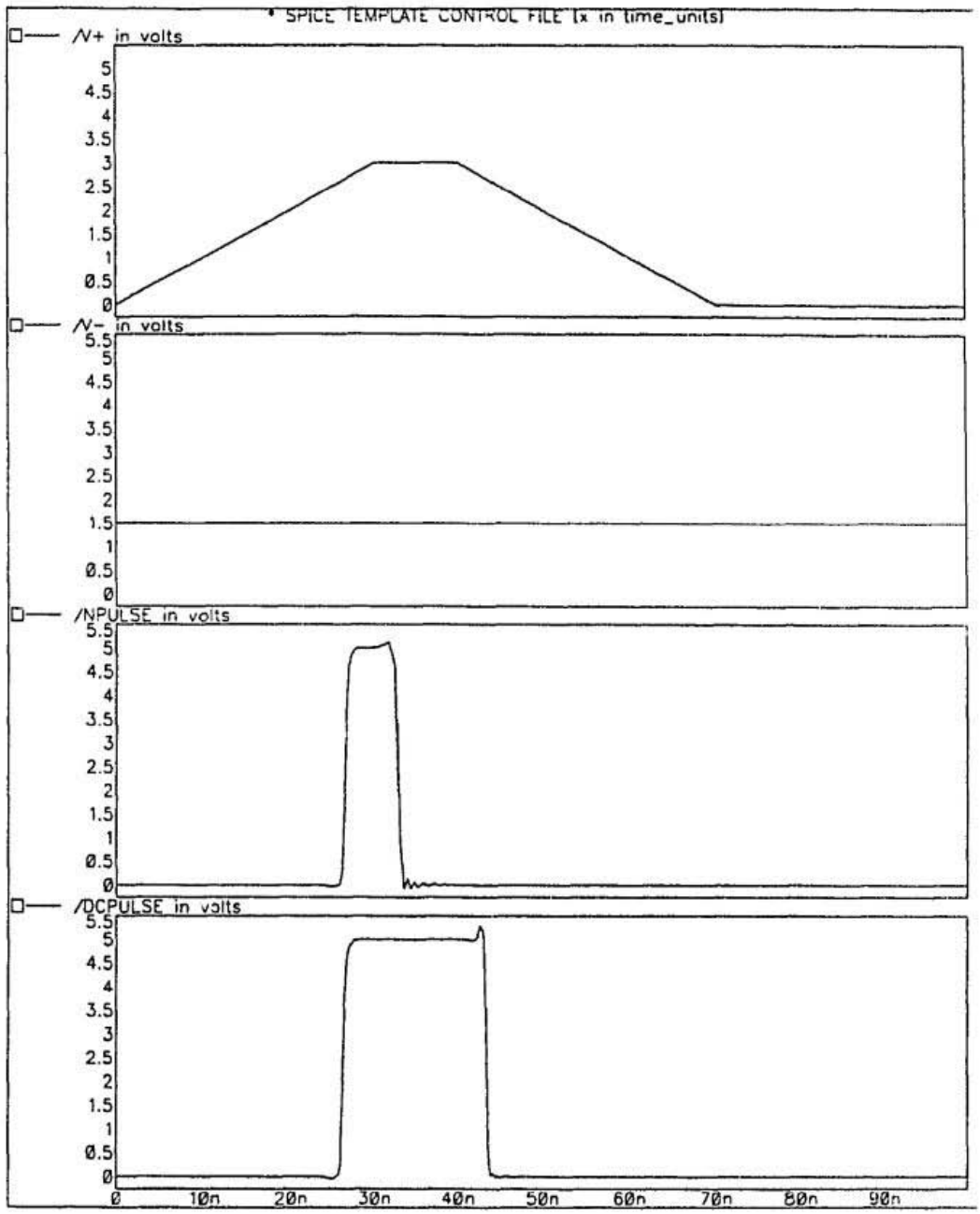

Figure 5.9: Spice simulation of the standard neuron. First two waveforms are input voltage and reference voltage. The next one is the output pulse while the last one is the discharge pulse.

5.4.3 Circuit Analysis

Analysis of the comparator and the associated delay is well documented [Allen et. al., 87], [Geiger et. al., 90]. However, the relevant portion of the design follows next.

Comparator

Referring to figure 5.8, under the balanced condition (Vg1=Vg2), I_5 splits equally between M1 and M2. So one needs M1 and M2 matched and similarly M3 and M4 matched. So, S1=S2 and S3=S4 where S (=W/L) is the shape factor and $I_1 = I_2 = I_5/2$. The drain voltage of M3 and M4 are equal and I_4 is mirrored to M6 by the ratio of M6 to M4. Similarly, I_5 is mirrored to the output by the ratio of M7 to M5.

$$\begin{aligned} I_7 &= I_5 * \frac{S7}{S5} \\ I_6 &= I_4 * \frac{S6}{S4} \end{aligned} \tag{5.23}$$

Since the circuit is balanced, $I_6 = I_7$ and $I_2 = I_4 = I_5/2$. Therefore, $2\frac{S7}{S5} = \frac{S6}{S4}$. In this particular circuit, $\frac{W1}{L1} = \frac{W2}{L2} = \frac{5.4\mu m}{3\mu m}$ and $\frac{W3}{L3} = \frac{W4}{L4} = \frac{10.8\mu m}{3\mu m}$. $\frac{W7}{L7} = \frac{5.4\mu m}{6\mu m}$ and $\frac{W5}{L5} = \frac{10.8\mu m}{3\mu m}$. This leads to $\frac{W6}{L6} = \frac{5.4\mu m}{3\mu m}$.

The propagation delay of the comparator can be estimated as follows. Since the delay is different for a rising and a falling output, the total rising delay (T_p^+) is the sum of the falling delay of the first stage (T_{p1}^-) and the rising delay of the second stage (T_{p2}^+). Similarly, the total falling delay (T_p^-) is the sum of the rising delay of the first stage (T_{p1}^+) and the falling delay (T_{p2}^-) of the second stage. Instead of getting into the detailed discussion (which can be found in [Geiger et. al., 90]), a brief outline is presented below.

$$T_{p1}^+ = C1\frac{V_{TRP2} - V_{OL1}}{I_5}$$

$$
\begin{aligned}
T_{p2}^{-} &= C2\frac{V_{OH2} - V_{OL2}}{2I_6} \\
T_{p1}^{-} &= C1\frac{V_{OH1} - V_{TRP2}}{I_5} \\
T_{p2}^{+} &= C2\frac{V_{OH2} - V_{OL2}}{2I_{6max}}
\end{aligned} \tag{5.24}
$$

where V_{TRP2} is the trip point of the second stage and is given by $V_{DD} - V_{t6} - \sqrt{\frac{2I_6}{\beta_6}}$ and V_{OH} and V_{OL} refer to the output high and low levels. I_{6max} is the maximum current available to charge the final load capacitor (C2) and C1 is the parasitic capacitance at the output of the first stage. For the above designed circuit, T_p^+ and T_p^- are found to be 4.5ns and 6ns whereas Spice simulation shows them to be 4.7ns and 5.6ns.

Delay in the pulse generators

The pulse widths of the neural circuit depends on the inverter delays for different load capacitors. The delay calculation for the inverter pair is well documented in the literatures [Mukherjee, 85], [Geiger et. al., 90]. The total delay (t_d) is given by the sum of the delays due to high to low transition (t_{HL}) and low to high transition (t_{LH}). The transistors used in the chain of inverters are of minimum sizes. t_d can be expressed in terms of the characteristic time constant for the process (depends only on the geometrical and electrical parameters but not on any particular circuit), τ_p by the relation $t_d = 8\tau_p$. τ_p is given by

$$
\tau_p = \frac{L}{K'W(V_{DD} - V_{tn})}C_{gate} \tag{5.25}
$$

The value of τ_p for the parasitic load is 0.2 ns and is 0.52ns for both parasitic and external capacitor (0.1 pF). So the approximate delay or the pulse width of the neuron output is (0.2*4*4 + .5*4) or 5.2 ns. Similarly the discharge pulse width is (.2*4*10 + .5*4*3) or 14ns. The delay due to the final stage nand - inverter pair is not included in the calculation. Spice simulation shows these pulse widths to be 6.5 and 17ns.

5.5 Input Neurons

5.5.1 Circuit Description

As mentioned earlier, two types of input neurons have been designed. The standard input neuron fires at its maximum rate with an input of 5 volts and the inverting input neuron does the same for an input of 0 volt. The circuits are shown in figure 5.10 and figure 5.11. The neuron employs a constant current source and reflects the current to a capacitor through a current mirror. Transistor M1 is the constant current source which sinks or sources current depending on the gate control voltage Vc. This current is mirrored by M2 and M3 (and also by M4 and M5 for inverting input neuron) to the capacitor C_{in}. The capacitor output is connected to the standard neuron. When the capacitor voltage goes past the neuron threshold, one output pulse is generated. However, the capacitor is discharged by the buffered comparator output instead of the discharge pulse. Consequently, the discharge pulse generator portion of the standard neuron has been dropped from the circuit. This is required because initially the capacitor will be fully charged and the comparator output will be high. Since the pulse generator needs an edge to generate the pulse, no pulse will be generated. So the high output of the comparator will discharge the capacitor initially and subsequently, every time after the pulse is generated. The inherent delay of the comparator ensures that the capacitor will be fully discharged before the voltage can ramp up the capacitor in the next cycle. The width of the discharge transistor (M6) has been taken to be 12 μm so that discharge time is very small.

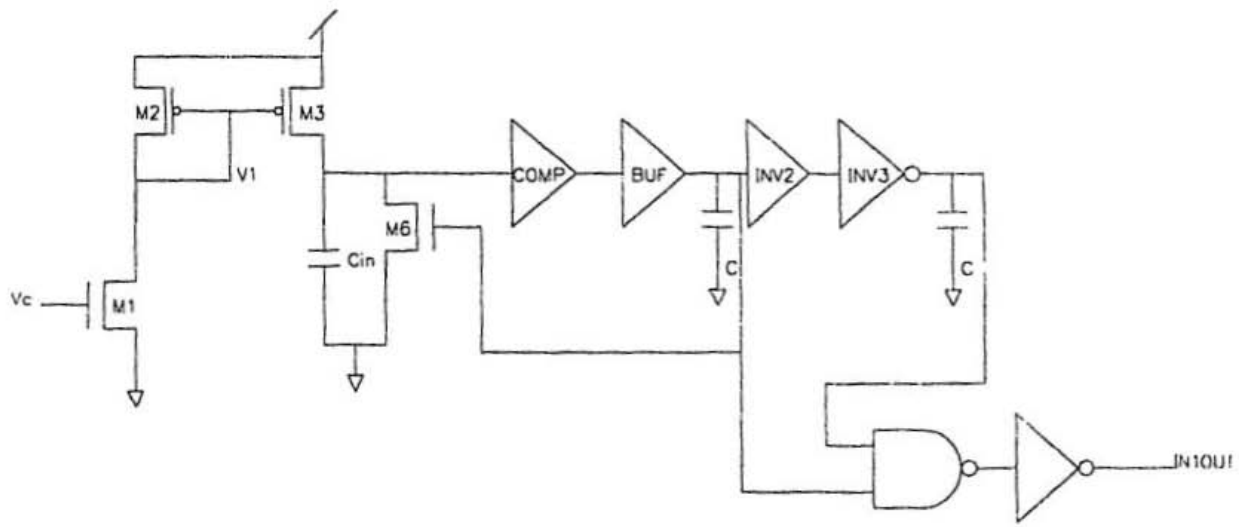

Figure 5.10: Schematic diagram of the standard input neuron.

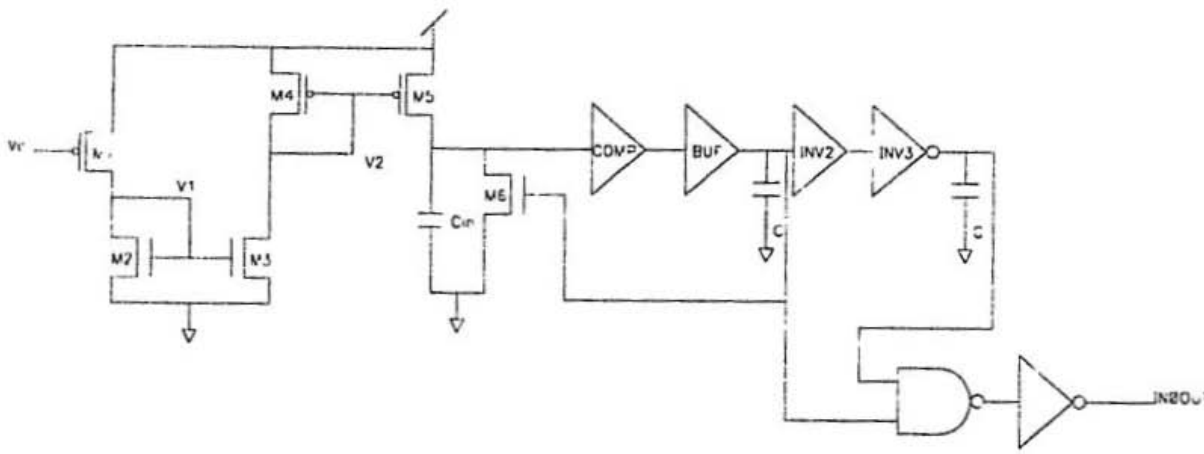

Figure 5.11: Schematic diagram of the inverting input neuron.

5.5.2 Design & Analysis of standard input neuron

It will be assumed that the drain of M1 is set to 1.5 volts. This will put M1 in the linear region and the current through it is given by

$$I_1 = \frac{K1'}{2} S1[2(Vc - V_{t1}) - V1]V1 \tag{5.26}$$

where $V1$ is the drain voltage of M1. M2 is in saturation and the current is given by

$$I_2 = \frac{K2'}{2} S2(V1 - V_{DD} - V_{t2})^2 \tag{5.27}$$

Solving for $I_1 = I_2$ it can be observed that to achieve $V1$=1.5 volts, $\frac{S2}{S1} = 4.87$. Setting $W1 = 5.4\mu m$, $L1 = 6\mu m$ and $L2 = 3\mu m$ one gets $W2 = 13\mu m$. The expected current is then 189 μA. But Spice level 3 simulation shows this current to be 110 μA. Investigating this discrepancy, it was found ([Vladimirescu et. al., 80]) that Spice in level 3 uses higher order effects as well as many empirical relations which are absent in the stated simplified equations. Surface mobility modulation by the gate voltage is given by

$$\mu_s = \frac{\mu_0}{1 + \theta(V_{gs} - V_{th})} \tag{5.28}$$

and is 526.14 cm^2/v-s. This is further reduced by the saturation of the hot electron velocity in the linear region reducing the effective mobility to

$$\mu_{eff} = \frac{\mu_s}{1 + \frac{\mu_s}{VMAX \cdot L} V_{ds}} \tag{5.29}$$

which is 465 cm^2/v-s. $\beta = (W/L)\mu_{eff} C_{ox}$ and is 2.89 $E^{-5} A/V^2$. Finally the drain current is given by

$$I_{ds} = \beta[V_{gs} - V_{th} - \frac{1 + F_B}{2} V_{ds}]V_{ds} \tag{5.30}$$

which comes to 150 μA accounting for only half the discrepancy..

In order to achieve a pulse train with a period of 30ns, the proper value of the capacitor has to be chosen. The pulse is generated 10.2ns after the threshold value is reached and comparator output goes low 13.3ns after the V+ input goes low. So, the capacitor has to be charged to 1.5 volts in (30-10.2-13.3) ns or in approximately 7ns. So, C_{in} is found to be 110e-6*7e-9/1.5 F or 0.5 pF. The capacitor will be charged to 3.78 volts when the comparator output goes high. The discharge current is computed to be 1.47 mA and it takes 0.8 ns to discharge the capacitor. The comparator takes another 13.3 ns to get the output low. However, the total time required for the comparator to make the output high is 7+10.2 ns or 17.2 ns. This ensures that the comparator output will go all the way to a logic low. The circuit has been simulated by Hspice and the different periods obtained for different control voltages are noted below. The simulation of the above circuit

$V_c(volts)$	Periods(ns)
5	30
4	33
3	41
2	68
1	338
0	∞

Table 5.1: Control voltages and the periods of the generated pulses for the standard input neuron.

for a control voltage of 5 volts is shown in figure 5.12.

5.5.3 Design & Analysis of Inverting Input Neuron

The following design procedure is for a control voltage of 0 volt. V1 will be assumed to be 1.5 volts. Then transistor M1 is in linear region and M2 is in

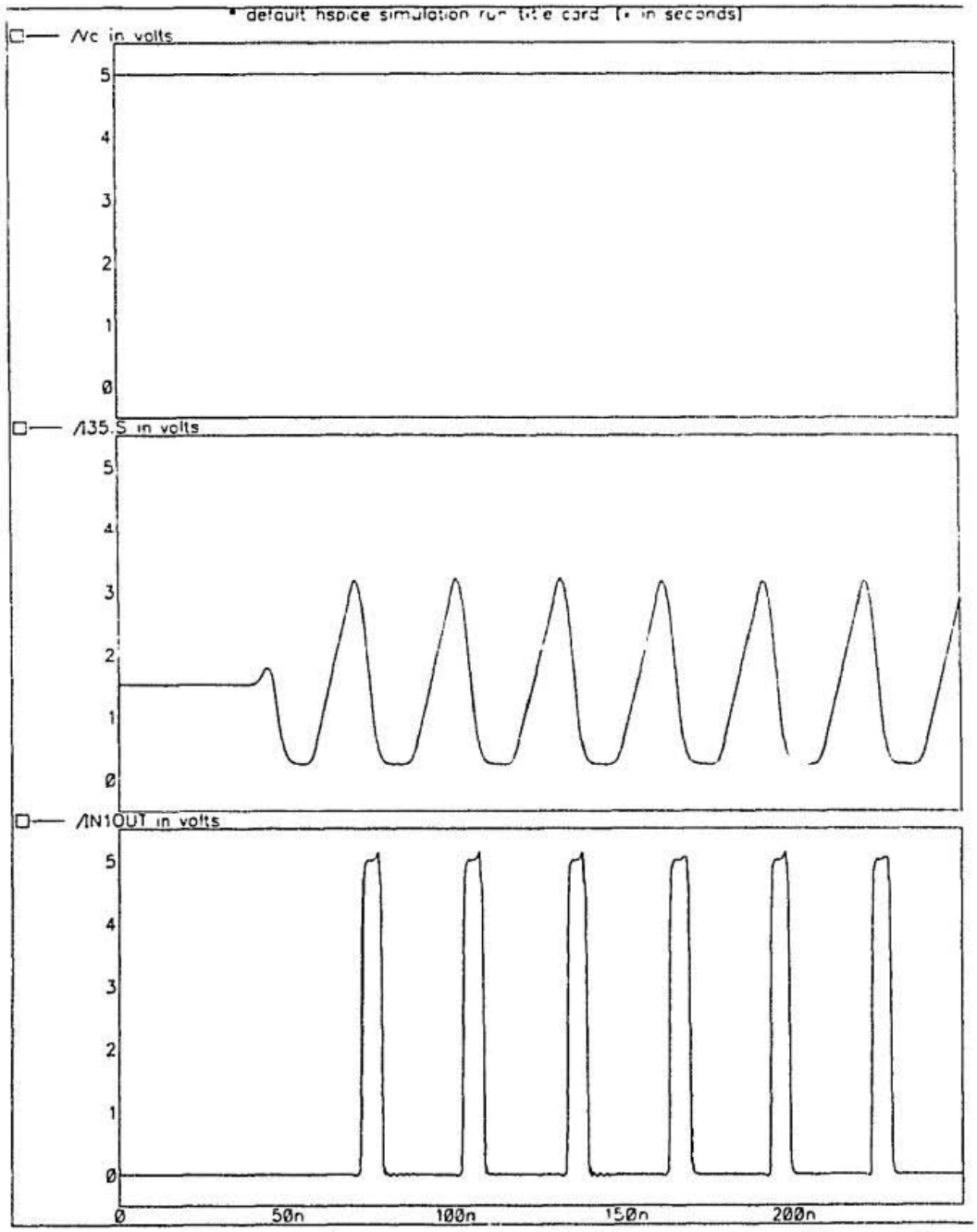

Figure 5.12: Hspice simulation of the standard input neuron. The waveforms are for the control voltage, voltage across the capacitor and the output pulses respectively.

saturation. The currents through M1 and M2 are given by

$$\begin{aligned} I_1 &= \frac{K'_P}{2} S1(V_c - V_{tp} - \frac{V_{DS}}{2})V_{DS} \\ I_2 &= \frac{K'_n}{2} S2(V1 - V_{tn})^2 \end{aligned} \tag{5.31}$$

Equating I_1 and I_2 and substituting values of all other parameters, one gets $\frac{S2}{S1}$=4. W1 is chosen to be 5.4μm and L1 is 6μm and this leads to W2 = 10.8μm if L2 = 3μm. S3 is taken to be equal to S2. If V2 is set to 1.5 volts, then both M3 and M4 are in saturation. Equating the currents results in S4 = 1.1 and consequently W4 = W5 = 5.4μm and L4 = L5 = 4.9μm. The charging current is calculated and simulated to be 46μA. Using the same logic as in the previous subsection, C_{in} is found to be 0.22 pF. Most of the other discussions for the standard input neuron also hold here. Time periods of the output pulses for different control voltages are given below. The simulation with Hspice for a control voltage of 0 volt is shown

$V_c(volts)$	Periods(ns)
0	30
1	34
2	44
3	80
4	863
5	∞

Table 5.2: Control voltages and the periods of the generated pulses for the inverting input neuron.

in the figure 5.13.

5.6 Concluding remarks

In this chapter, design procedure of the neural circuits have been discussed. Detailed mathematical analysis has also been included. Each of the circuits has

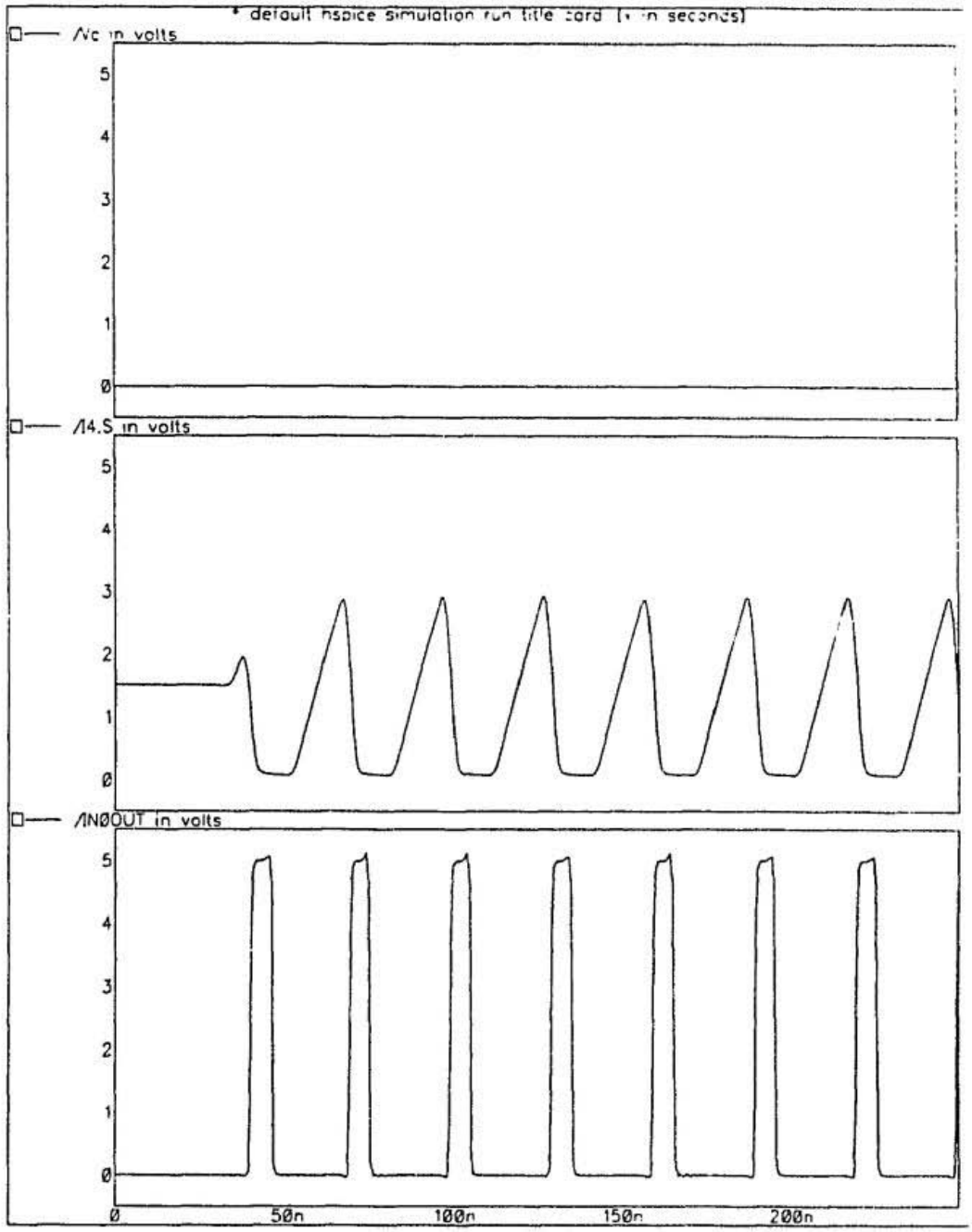

Figure 5.13: Hspice simulation of the inverting input neuron. The first waveform is the control voltage, then the voltage across the capacitor and finally the output pulses.

been simulated using Spice and the result has been compared with the theoretically calculated values. Equations 5.13, 5.16, 5.17, 5.19, 5.20 and 5.22 can be used to develop a simulator which is needed for simulating a large network. This is required because Spice is not very suitable for simulating a large network. Instead of getting into the detailed circuit equations, these equations can be used to generate a macro model that can be used very effectively to simulate and study a large network.

Chapter 6

Standard Cells

6.1 Introduction

In developing an integrated circuit, a top down design approach is usually taken. The whole circuit consists of a few big functional blocks which are decomposed into subblocks in the next lower level of hierarchy. This is continued until the transistor level is reached. The same idea is also used for the physical design. Different blocks or cells are developed which are functionally unique and are connected together to realize the full circuit.

Working on the cell level has several advantages [Lauther, 83]. Once one cell is designed, it can be used repeatedly in the same circuit leading to a high degree of regularity of the layout. This reduces design time and verification expense significantly. Once the cell is working properly, it can be used in other designs without any further expense. Not only that, the number of components with which one has to deal (manually or using place and route routines) is much less compared to the transistor level approach. Finally the standard cell approach leads to a very structured design. Functional details are hidden in the cell and one has to bother only with the input and output pins.

The standard cells for the neural circuits have been developed using the Ca-

dence EDGE and is compatible with Canadian Microelectronic Corporation's CMOS3 DLM process. This is a $3\mu m$ CMOS technology with double metal layers and a single poly layer. The library was designed using the grids even though EDGE can do gridless routing [CMC. 89].

All the dimensions mentioned in this chapter are design scale microns (dsm). This is related to the actual or the physical micron by the following relation

$$\text{Physical micron} = \tfrac{3}{5} \text{ design scale micron}$$

6.2 Cell Specifications

- Grid size : each grid is $16\mu m$ wide with 16 subdivisions of one micron each. The grid size stems from the gap that should be maintained between two vias.
- Cell dimensions
  - Cell height : each cell has height of 7 grids or 112 dsms. The center of the vdd bus makes the top boundary whereas the center of the ground bus makes the bottom boundary. P-guard and N-well layers may extend beyond the bottom boundary.
  - Cell width : the cell width can be anything. It has been found that the EDGE place and route routine does not need the cell width to be integral multiple of grid units. This is the most noticeable deviation from the CMC CMOS3DLM library and leads to a compact cell design.
  - Cell origin : This is the left most center point of the ground bus.
  - Cell boundary : this corresponds to the left and the right cell boundaries but is 2.5 dsm above and below the top and the bottom of the actual cell.

- Power Buses : vdd and the ground buses run across the top and the bottom of the cell to the full width. Metal1 is used for this purpose and the buses are 10 dsm wide. 5 dsm metal is extended beyond the top and the bottom of the cell. Two metal1 pins (vdd! and gnd!) are placed on the vdd and the ground buses and their access direction is to the left and to the right.

- Input-output ports : All input-output ports are provided with metal1-via-metal2 pads. They are 11 dsm square and are placed in the cell at any position that was found convenient and area efficient. Ports originating from metal1 have metal1-via-metal2 pads whereas those from poly have poly-contact-metal1 and then metal1-via-metal2 pads. Most of the vias have 11 dsm metal2 pins on top with top and bottom access directions. In order to save area, some of the vias could not be provided with free top and bottom access. In that case, metal2 wire has been drawn to a convenient place with a 5 dsm square metal2 pin at the tip.

  Some of the cells are compound cells in the sense that they are composed of base cells. Some of the ports of the base cells which are meant for internal butting are provided with poly-contact-metal1 pads instead of vias. All the input and output ports of the compound cells have via connection.

- Port name : ports have been named in such a way that they give good indication about the incoming or outgoing signals (e.g. V_{ex} for excitatory signal, DCPULSE for discharge pulse etc.).

- Interior of the cells : except for the power buses that run across the full width of the cell, all metal1 and metal2 wires are at least 2.5 dsm from the left and the right boundaries. Poly wires maintain the same gap on all four

sides. The P-well can occupy the lower half of the cell and extends to the full width of the cell. This enables a continuous P-well along with the butting cells. The P-guard and the N-well can extend beyond the left, the bottom and the right boundaries. N+ and P+ layers are at least 2.5 dsm away from the boundaries whereas the N+ and P+ diffusions are at least 4 dsm away from all four sides. Care has been taken so that the design rule for the gap between the P-well and the diffusion are not violated.

6.3 Cell Description

The cells have been designed in such a way that they can be butted together to form a row of cells and the routing channels can be formed in between the rows. It is also possible to stack rows together (with alternate rows flipped upside down) without channels between them. This leads to a very compact layout. Metal2 wires run vertically upwards into the channels and the metal1 wires run horizontally. Except for the compound cells, use of metal2 wires has been restricted only on to the vias. For the compound cells, metal2 has been used for internal connections (when necessary) but this does not hamper the vertical access to the actual input output ports.

6.3.1 Excitatory Synapse

The layout of the excitatory synapse is shown in figure 6.1.

Dimension : 112 x 96 dsm

Input ports : $V_{ex}, V_{out}, V_{dc}, V_{lk}$

Output port : V_m

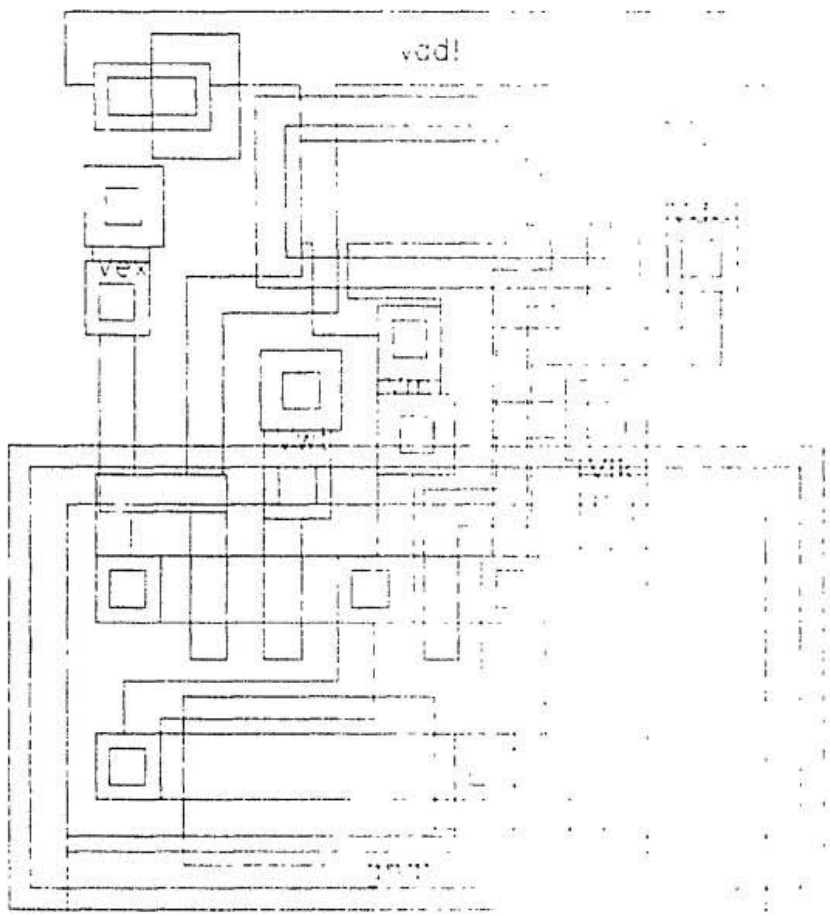

Figure 6.1: Layout of the excitatory synapse

6.3.2 Inhibitory Synapse

The layout is shown in figure 6.2.

Dimension : 112 x 48 dsm

Input ports : V_{in}, V_{wt}

Output port : V_m

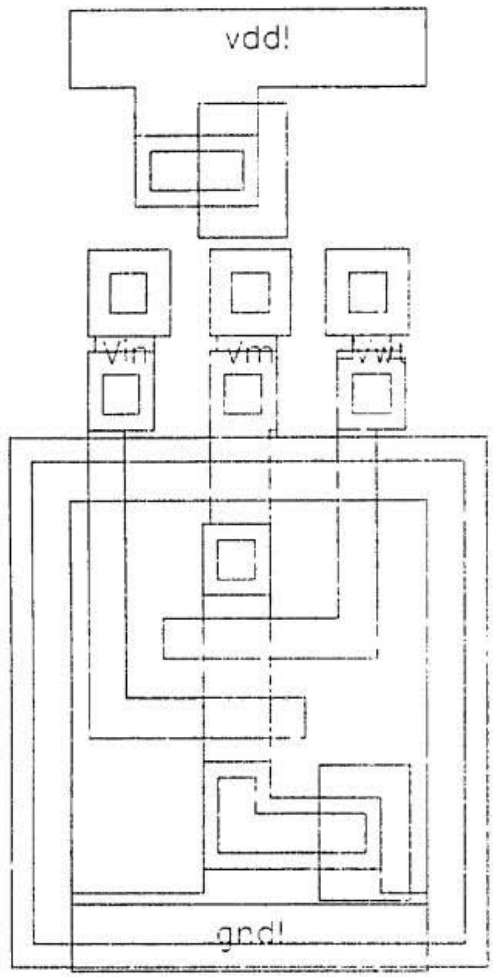

Figure 6.2: Layout of the inhibitory synapse

6.3.3 Standard Neuron

The standard neuron is a compound cell and is composed of the following base cells : comparator, buffer, delay (capacitor) elements, inverter2 (two minimum sized inverters in a row) and inverter3 (three minimum sized inverters). The layout of the neuron is shown in figure 6.3. The base cells are shown in figures 6.4 to 6.9.

Dimension : 112 x 901 dsm

Input ports : V+, V-, V_b

Output port : NPULSE, DCPULSE

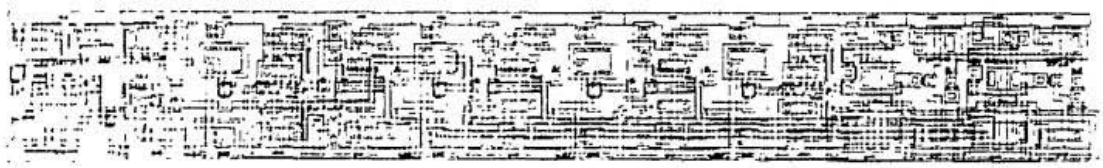

Figure 6.3: Layout of the standard neuron

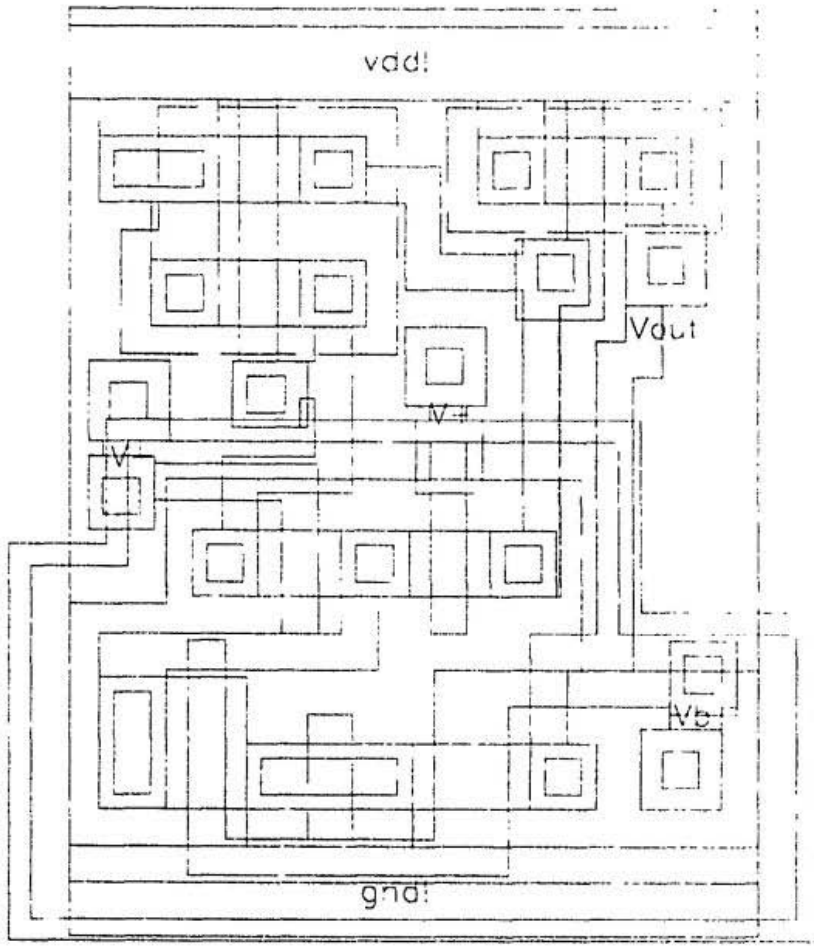

Figure 6.4: Layout of the comparator.

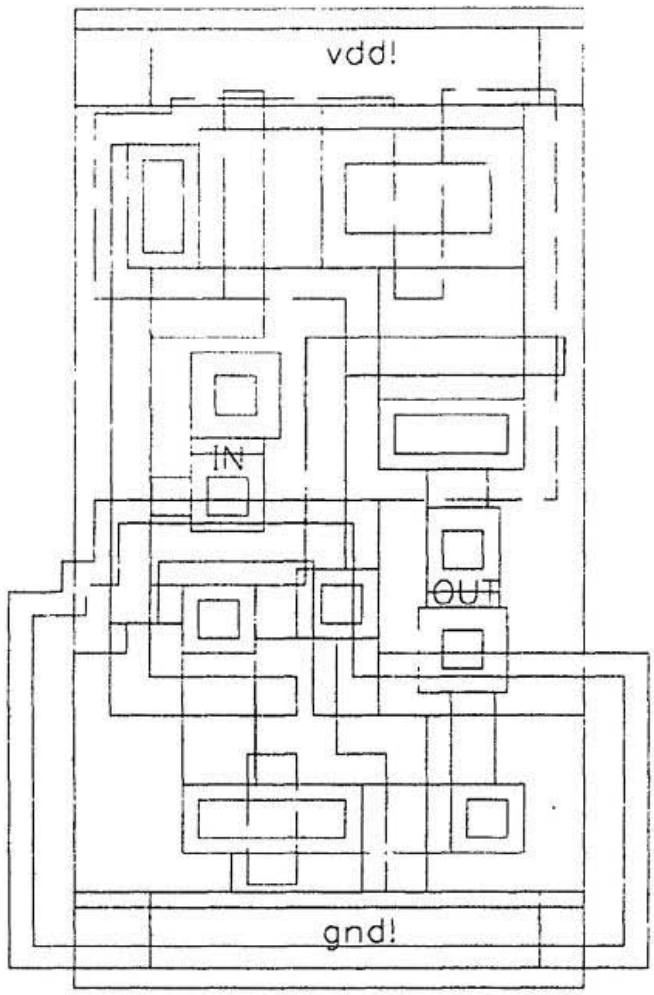

Figure 6.5: Layout of the buffer

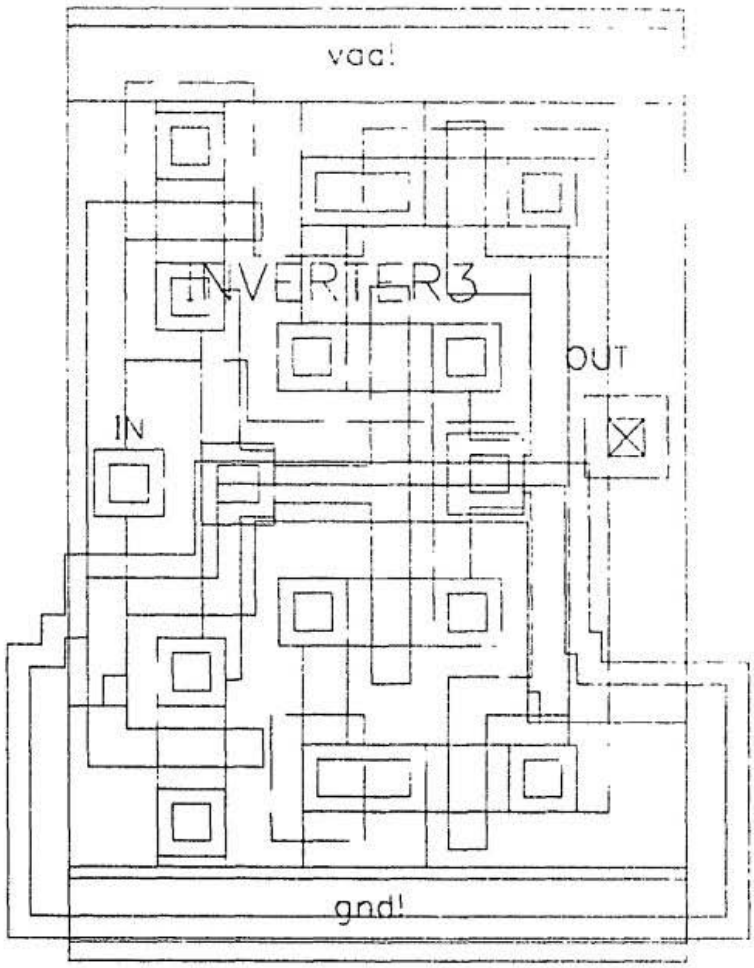

Figure 6.6: Layout of the inverter3.

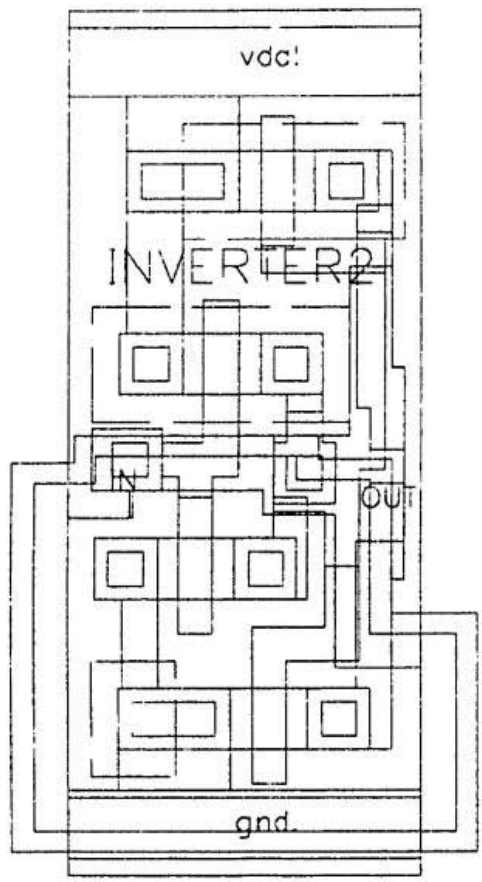

Figure 6.7: Layout of the inverter2.

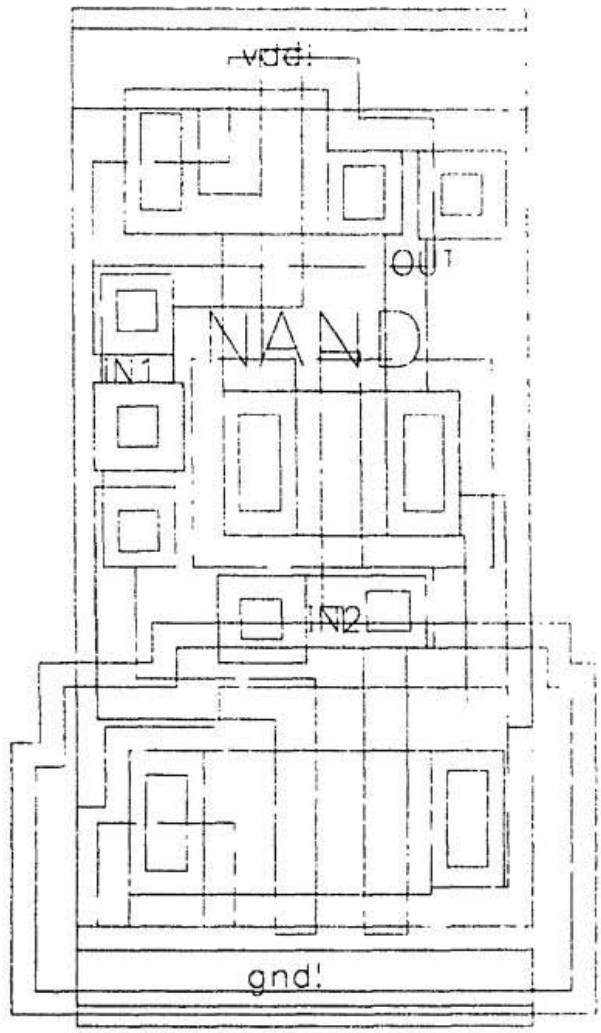

Figure 6.8: Layout of the two input nand gate.

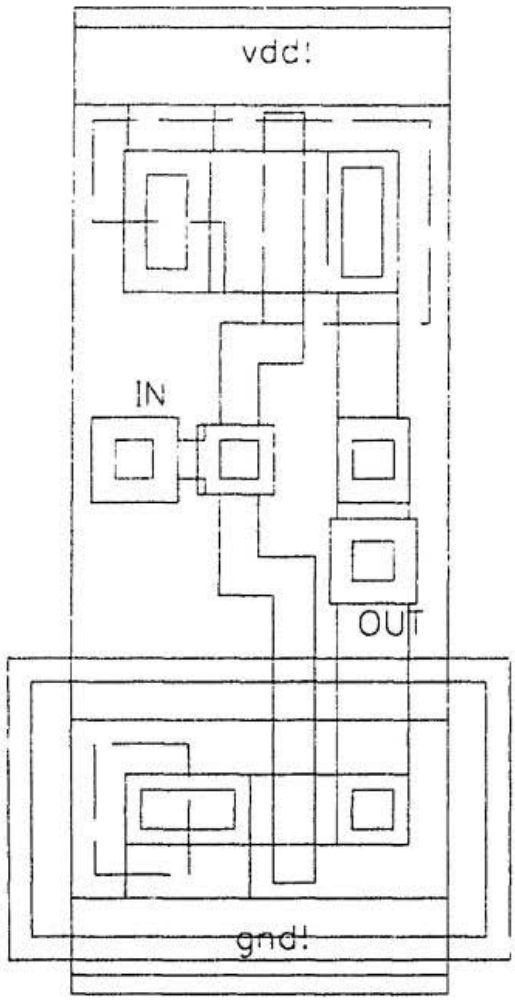

Figure 6.9: Layout of the inverter.

6.3.4 Inverting input neuron

The inverting input neuron is also a compound cell and is composed of inN0 (voltage dependent ramp generator), comparator, buffer, delay, inverter2 and inverter3. The layout of inN0 and the inverting input neuron are given in figure 6.10 and figure 6.11 .

Dimension : 112 x 595dsm

Input ports : $V_c, V-, V_b$

Output port : INOOUT

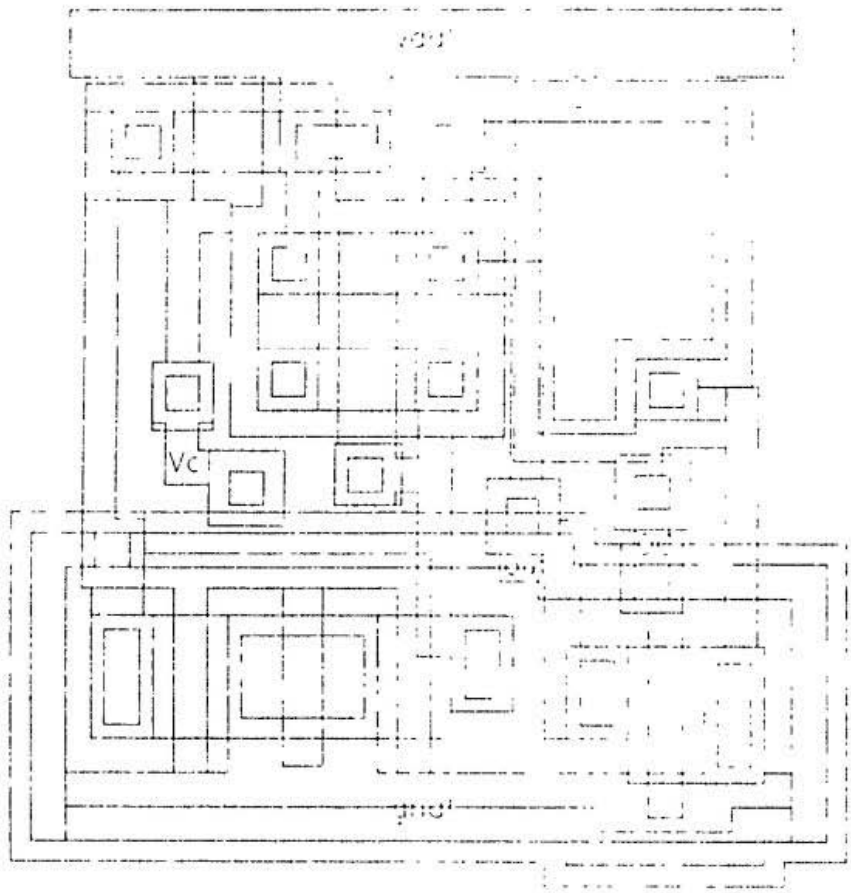

Figure 6.10: Layout of ramp generator inN0.

Figure 6.11: Layout of the inverting input neuron.

6.3.5 Standard input neuron

The standard input neuron is composed of inN1 (voltage dependent ramp generator, figure 6.12), comparator, buffer, delay, inverter2 and inverter3. The layout is shown in the figure 6.13. Dimension : 112 x 600dsm

Input ports : $V_c, V-, V_b$

Output port : IN1OUT

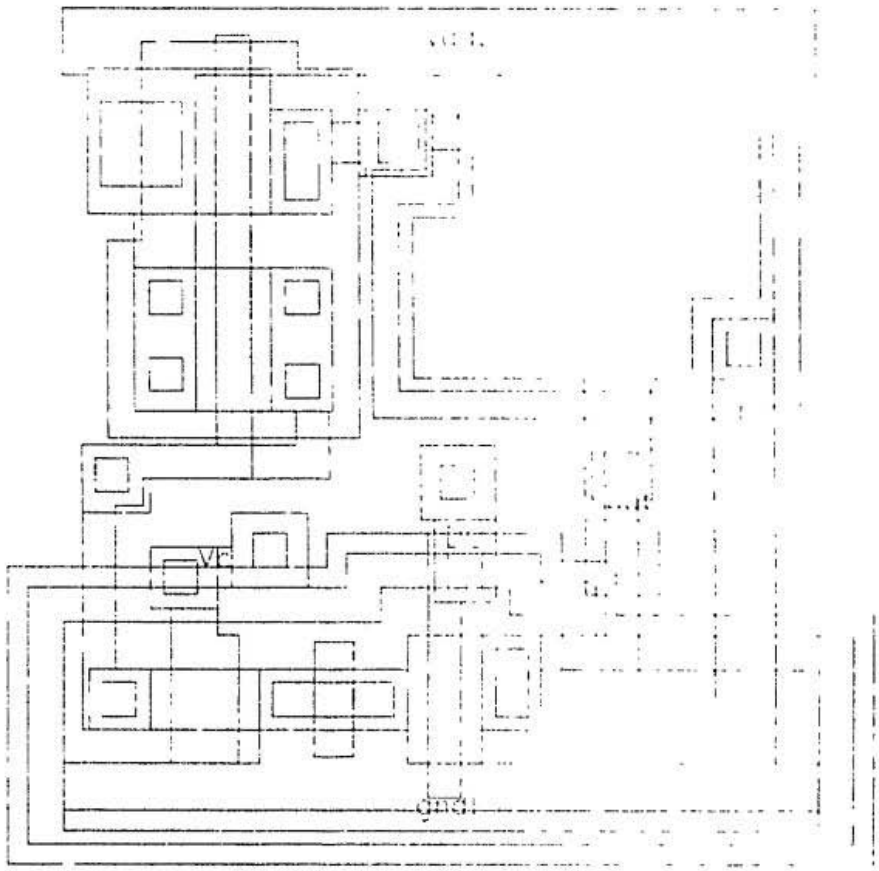

Figure 6.12: Layout of inN1.

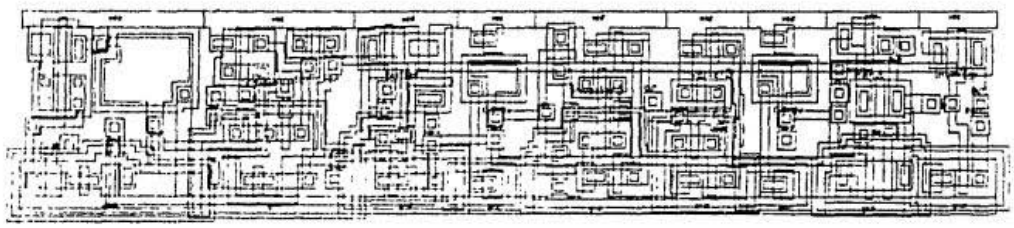

Figure 6.13: Layout of the standard input neuron.

6.4 Simulation

All the standard cells have been extracted and the simulations have been done on the extracted schematics (using Spice). Figure 6.14 and figure 6.15 show the simulations of the excitatory synapse and the standard neuron for the same kind of inputs as in the previous chapter (i.e. simulation on the schematics). The names of the waveforms are shown on the left of each of the waveforms. It can be observed that the result is somewhat different (compared to the simulation results on the schematics, figure 5.4 and 5.9) due to the presence of parasitic capacitances due to actual laying of different layers and routing between cells (in compound cells).

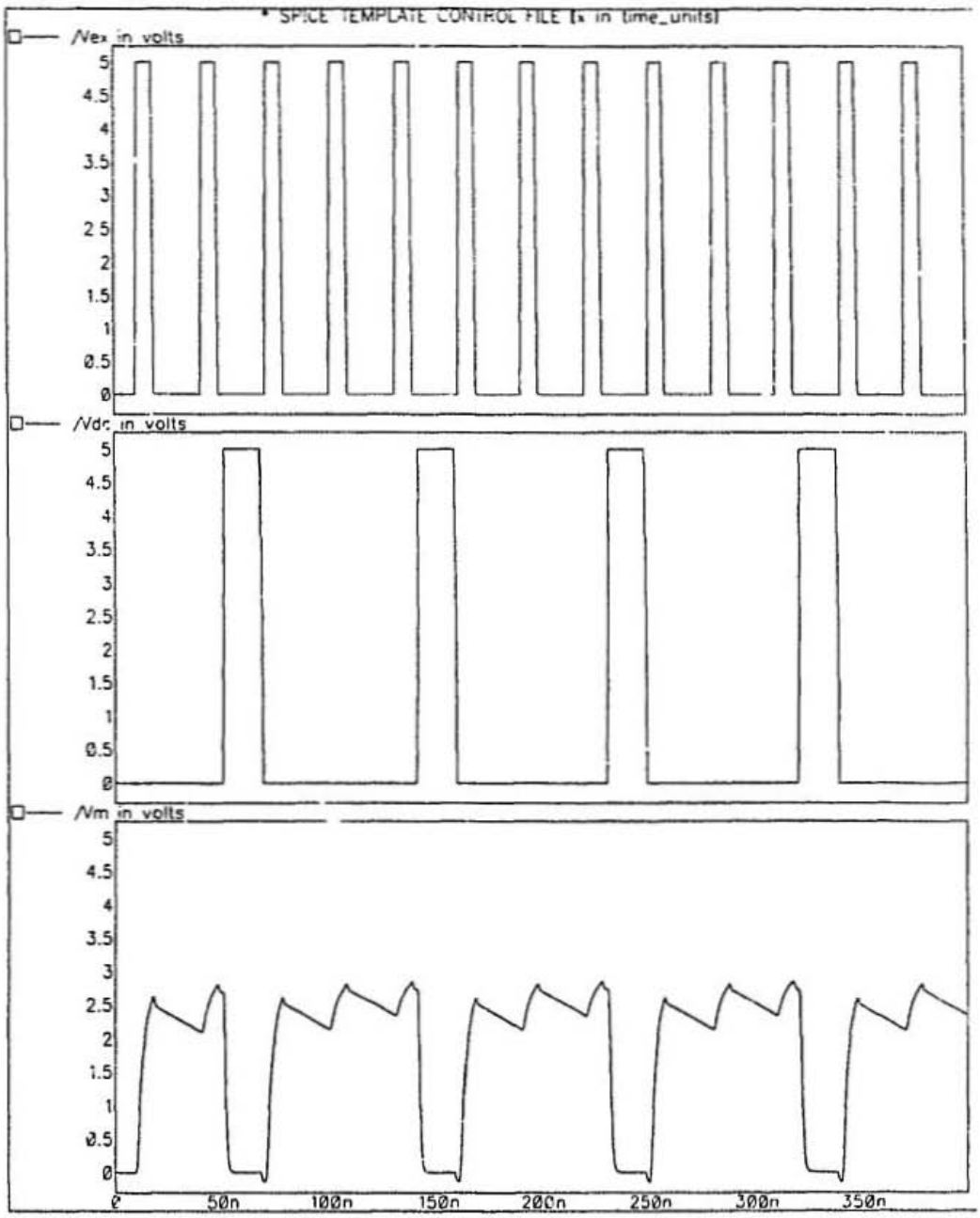

Figure 6.14: Simulation of the extracted layout of the excitatory synapse.

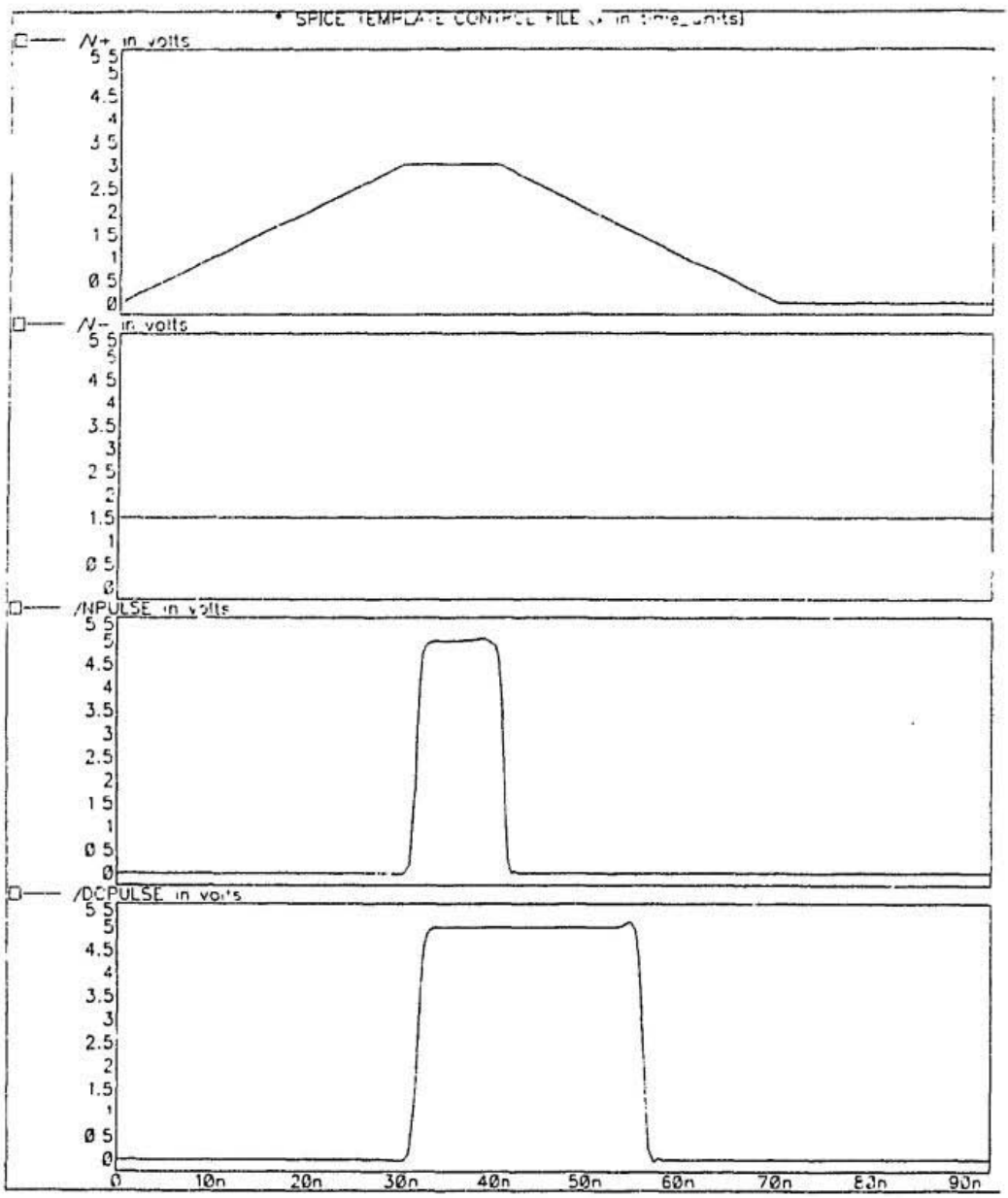

Figure 6.15: Simulation on the extracted schematic of the standard neuron.

Chapter 7

Simulations and Results

The proposed pulsed neural architecture is quite different from the existing ones and therefore needs some careful exploration. A number of networks such as pattern classifiers, associative memory, XOR gates, Hopfield nets etc. have been simulated using Spice or Hspice. The next few sections are devoted to some of these simulation results.

7.1 Pattern Classifier

Figure 7.1 shows a simple pattern classifier network which is basically a template matcher [Graf et. al. 88]. A number of vectors are stored in the network (here 7 vectors namely, 00000, 11111, 11110, 10101, 01010, 00100 and 11011, each 5 bits long). Input neurons feed each synapse in a row in parallel and the output of all the synapses in a column generate the activation voltage for the corresponding neuron. When an input vector is presented (00000 in this example), the network compares the vector with all the stored ones in parallel and generates the outputs. Figure 7.2 and 7.3 show the activation and the output of all seven neurons. It can be seen that the output firing rate depends on how closely the stored vector matches the input one. Thus, the network not only finds a match, but also indicates the Hamming distance. In this example, the network could successfully

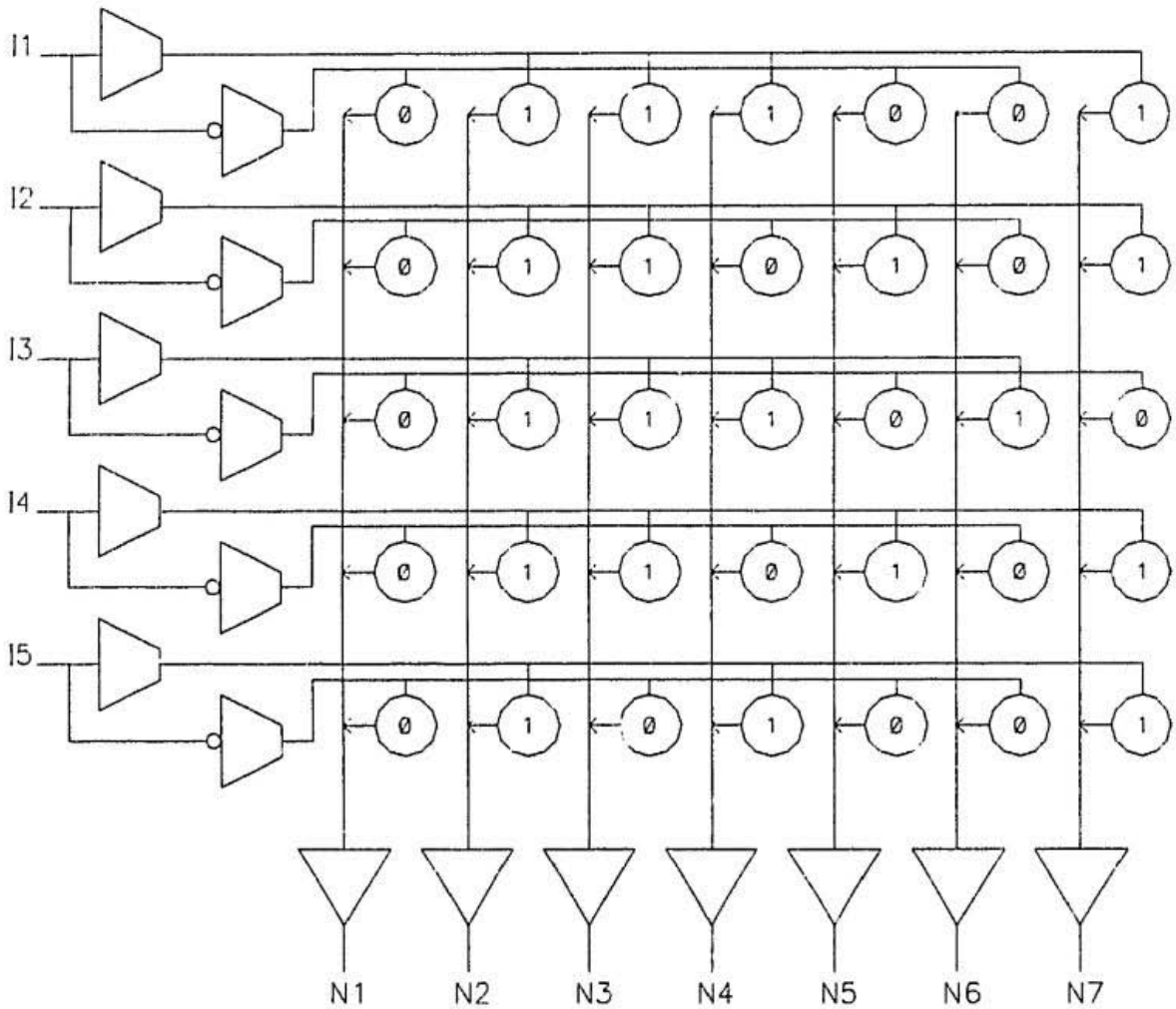

Figure 7.1: Template matching example. All weights are 3.6 volts. Stored pattern (0 or 1) is written inside the synapse.

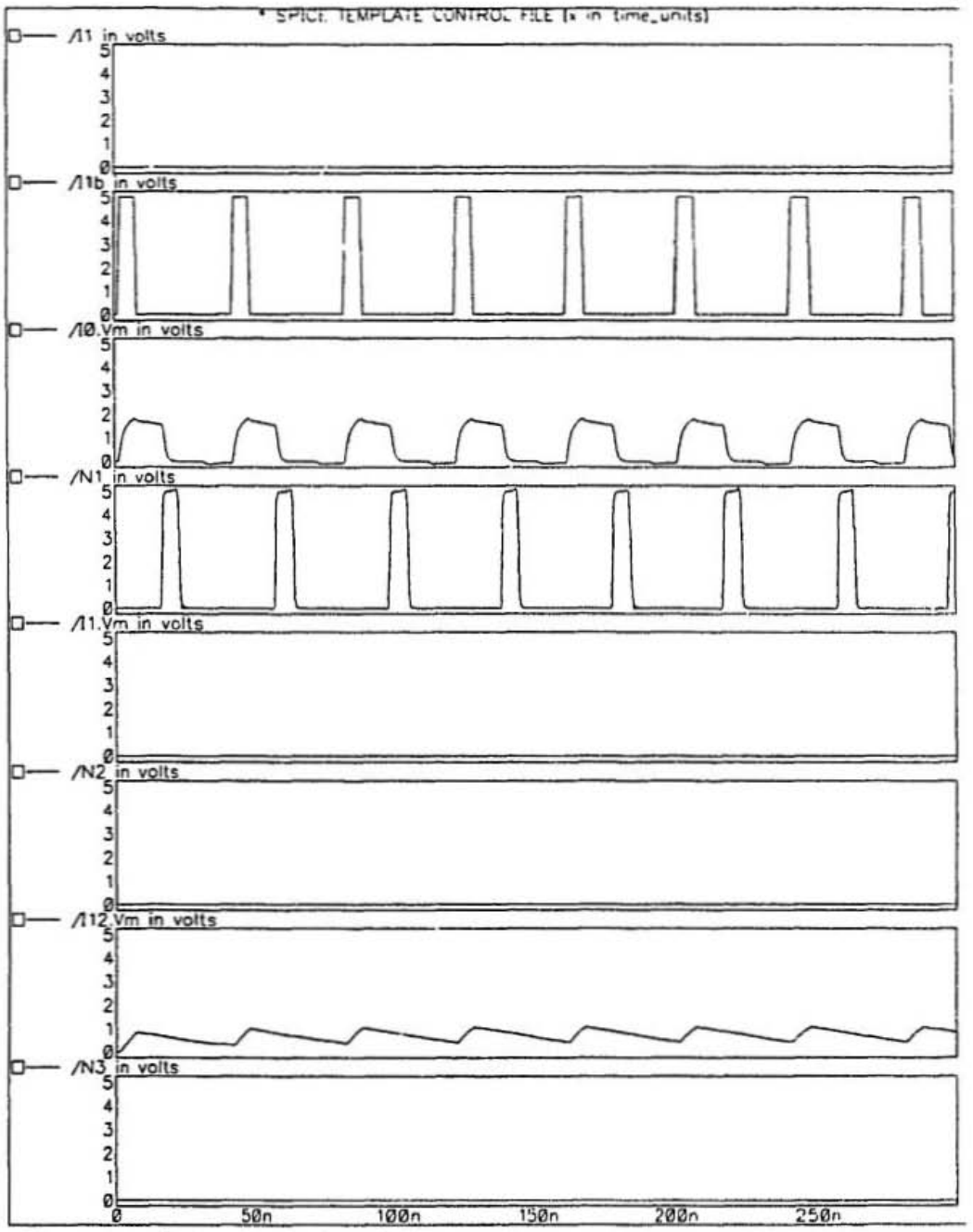

Figure 7.2: Spice simulation of the template matching example. First two waveforms indicate the outputs of two different input neurons. Rest of the waveforms are the activation and output of neurons 1, 2 and 3.

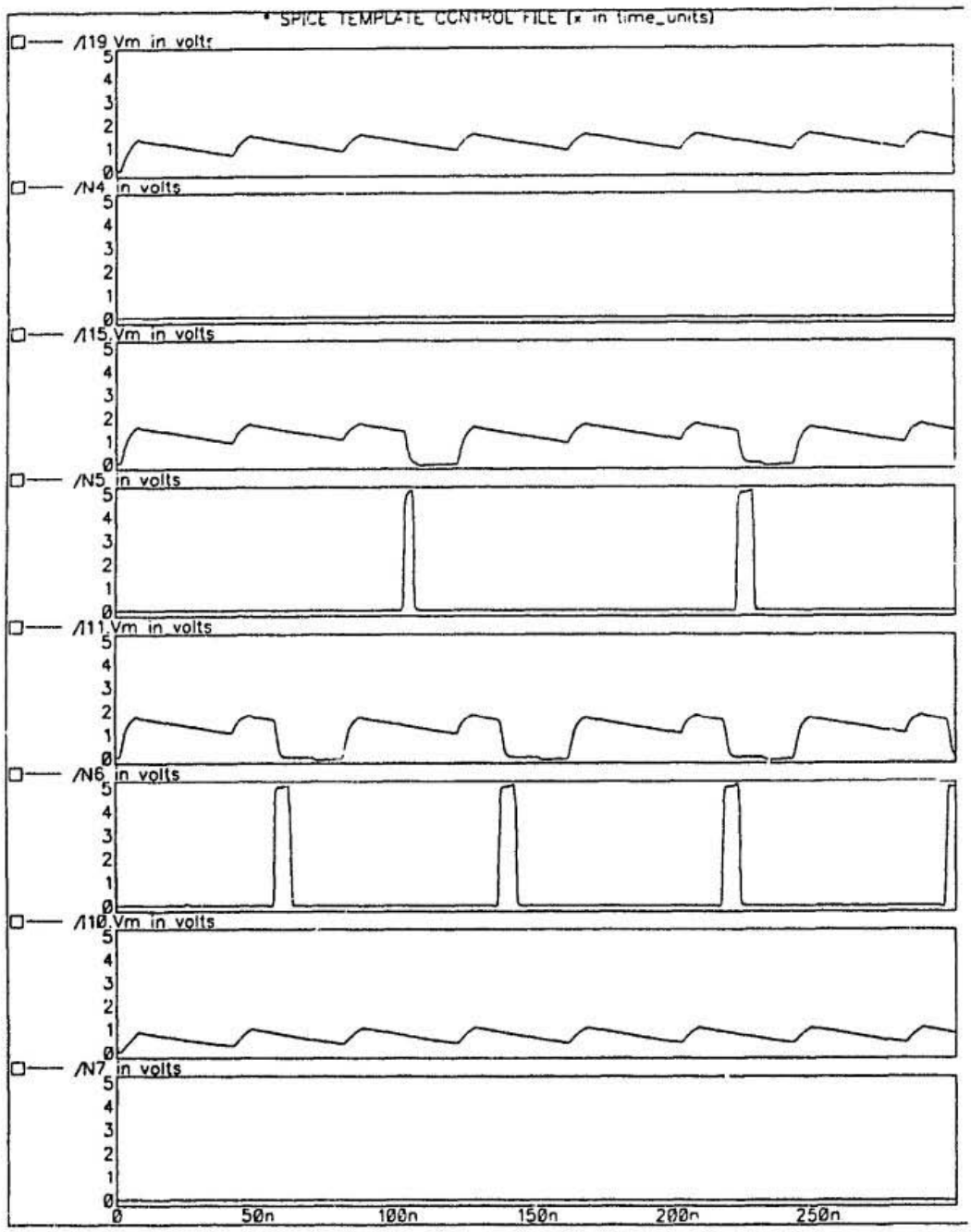

Figure 7.3: Spice simulation of the template matching example. The waveforms show the activation and output of neurons 4, 5, 6 and 7.

classify patterns which are perfectly matched or away by a Hamming distance of one or two. The network does not respond to a pattern away by a Hamming distance of three or more. This can be changed by changing the weight voltage or the threshold voltage of the neuron.

Figure 7.4 and 7.5 also give the simulation results of the network when probed by a vector (01100) not stored in the network at all. It is apparent that even if there is no exact match, the network can classify the stored pattern according to the Hamming distance.

In a pattern classifier, 1 and 0 components of the stored vectors are normally realized by excitatory and inhibitory synapses. This scheme is problematic for the proposed pulsed neural circuit because the total input activation of any neuron is essentially the number of ones minus the number of zeroes of the stored pattern. If the number of stored zeroes is more than the number of stored ones, there is no net activation and hence the neuron would be unable to detect the pattern. Most of the neural networks [Graf et. al., 88], [Hopfield, 82] use an inverted output of the neuron for the inhibitory synapse. However, the biological neurons use the same polarity signals for both excitatory and inhibitory synapses. This is the reason for creating the inverting input neuron which fires at a high rate for an input of zero and therefore works as a zero detector. By using both of these input neurons, one can store the patterns by using excitatory synapses only. All the synapses were driven by a common weight which was varied for optimum performance (the only form of learning available to us at present).

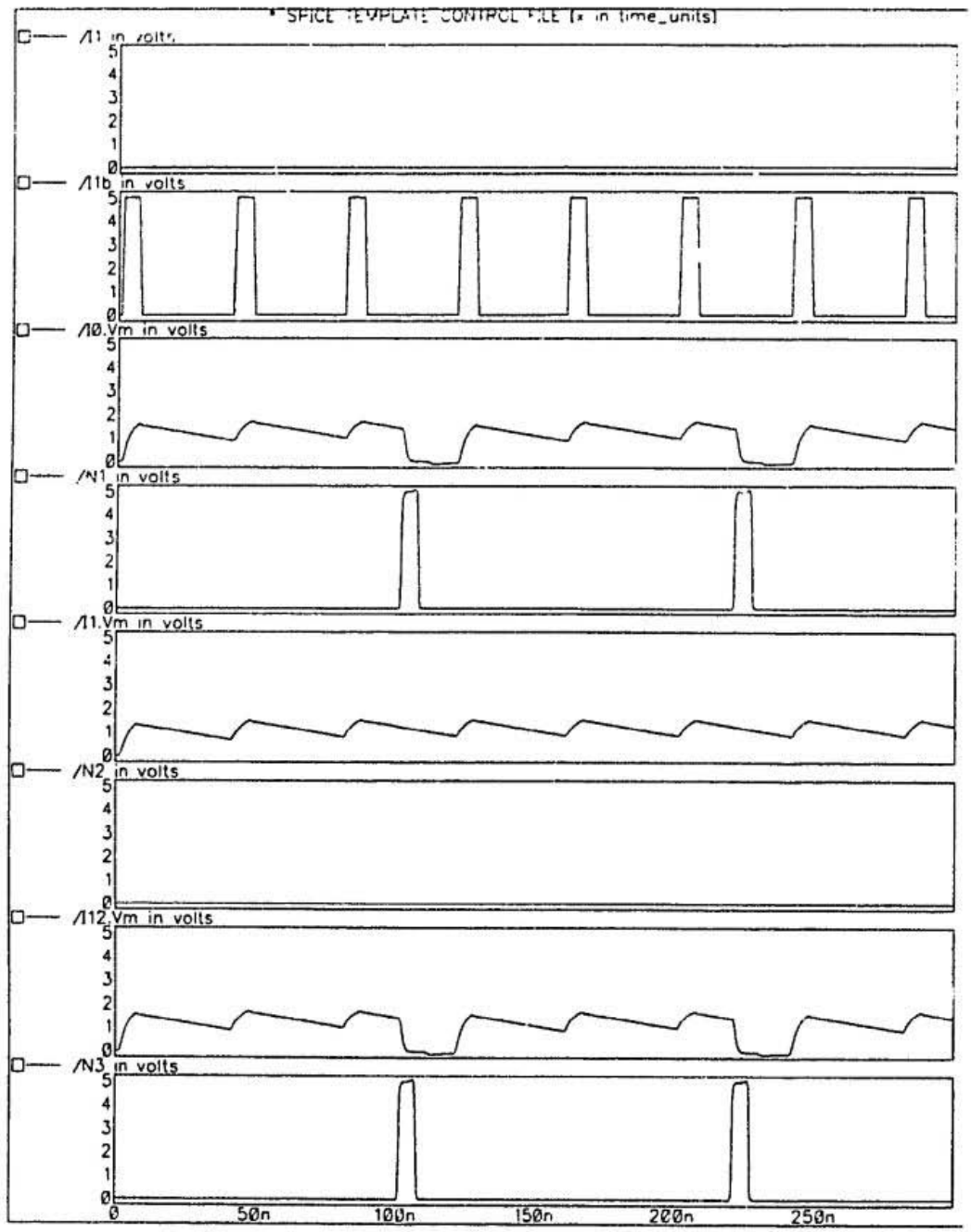

Figure 7.4: Spice simulation of the template matching example when probed with 01100. The order of waveforms are same as in figure 7.2.

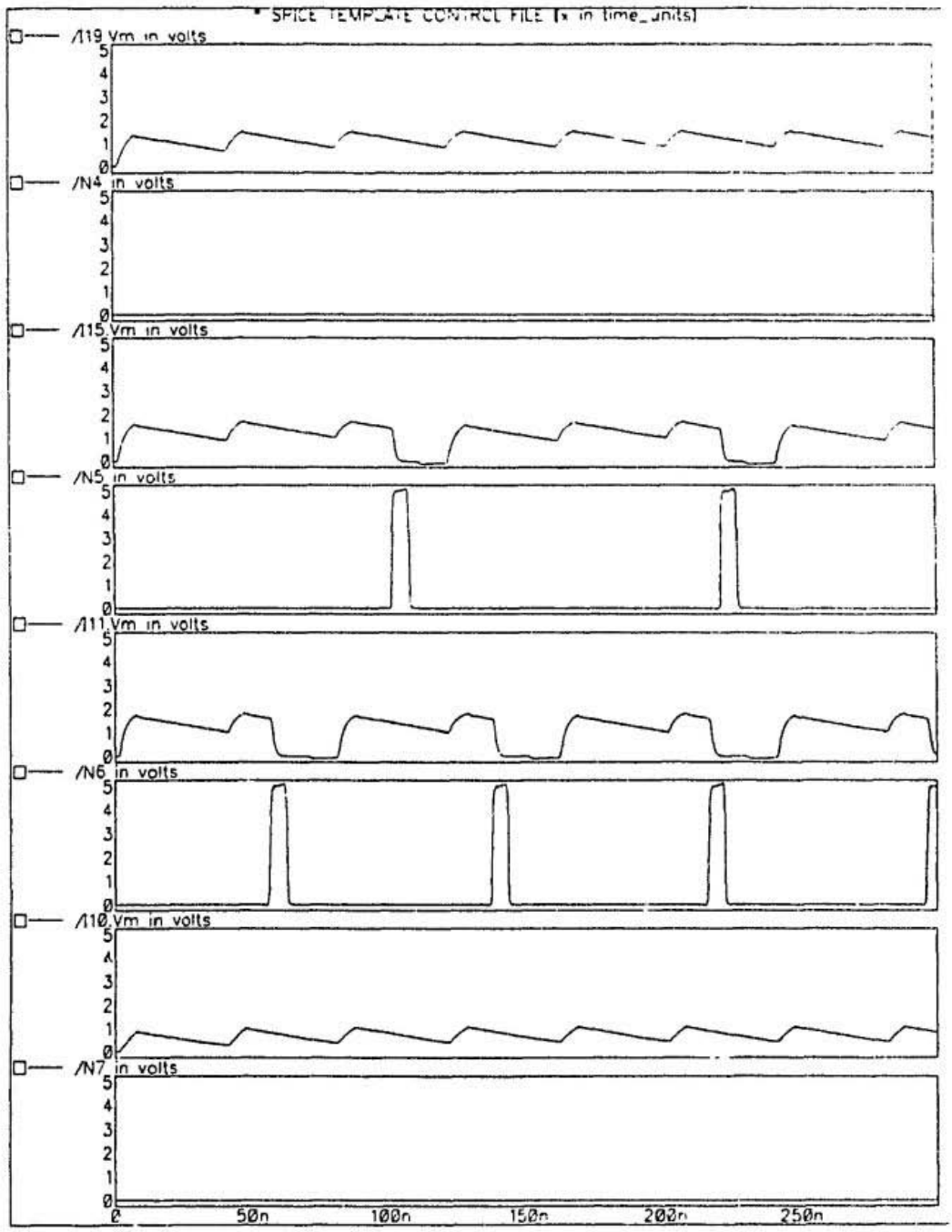

Figure 7.5: Spice simulation of the template matching example when probed with 01100. The same waveforms as in figure 7.3 are shown here.

A content addressable memory can be formed by a template matching network followed by a winner-take-all network [Graf et. al., 89]. The Hamming classifier or the template matcher finds out the overlap between the input vector and the stored ones and the winner-take-all network retrieves the pattern with maximum overlap. Figure 7.6 shows such a network with the Hspice simulation in figure 7.7. The output of each of the neuron inhibits all the other neurons. In the simulation, the input pattern is 01100 which closely matches (but not exactly) the sixth stored vector (00100). It can be seen that the network is able to detect the stored vector (neuron 6 is firing) properly. The firing rate is less due to the fact that the stored vector (00100) does not exactly match the probed one (01100). This can be taken care of by increasing the weight.

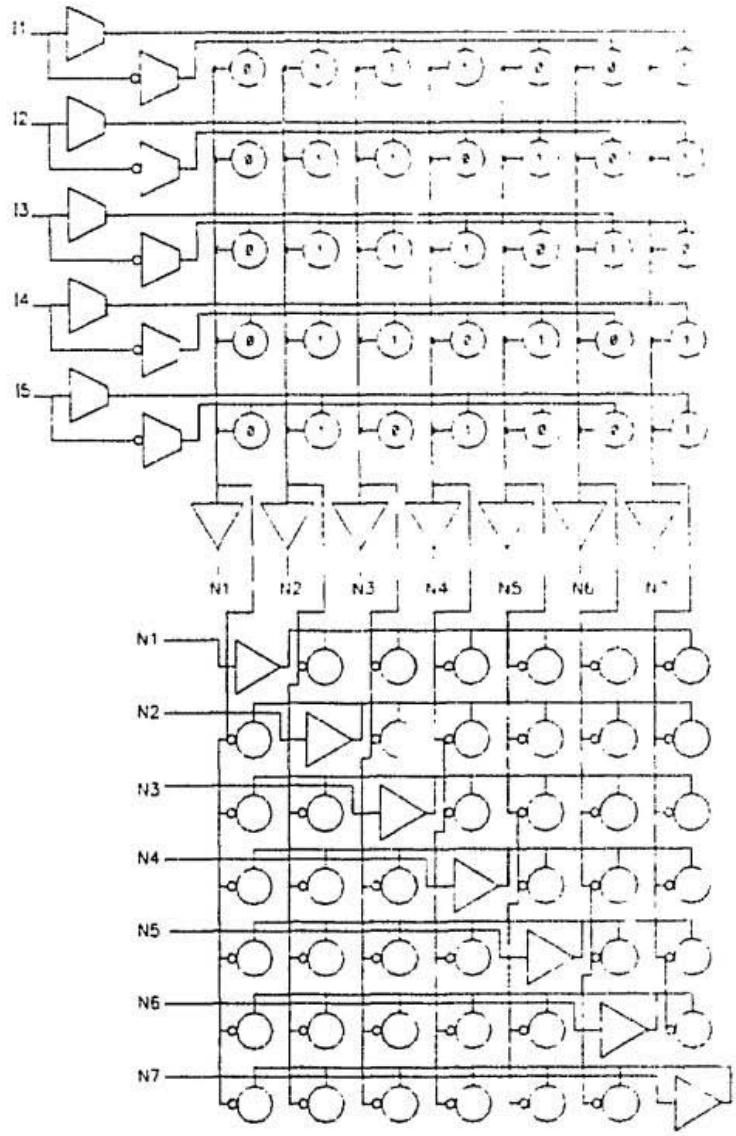

Figure 7.6: Content addressable memory formed by a template matcher followed by a winner-take-all network.

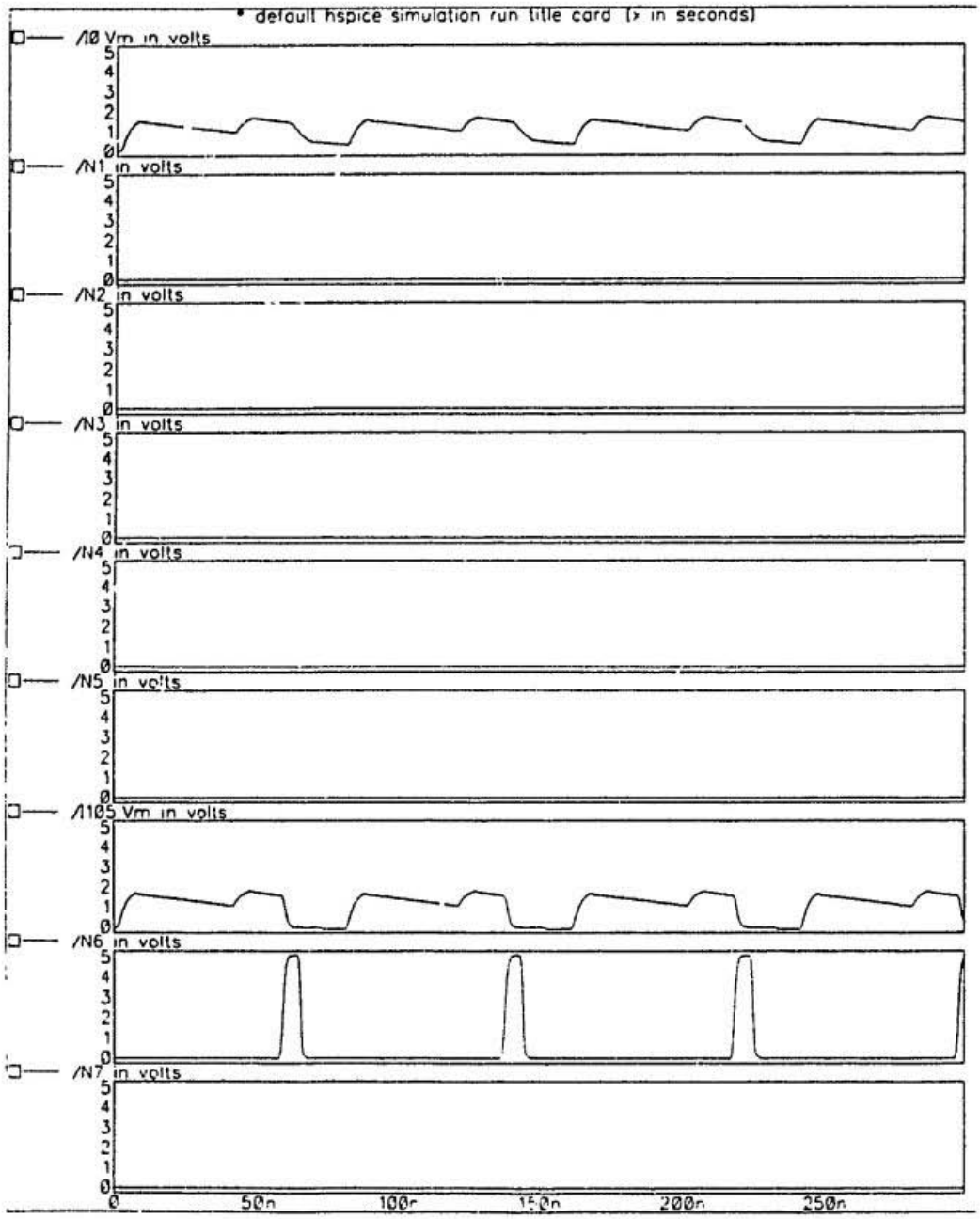

Figure 7.7: Output of 7 neurons (including the activation of neuron 1 and 6) when the presented pattern is 01100.

7.2 XOR Gate

Three different types of XOR gates have been simulated all of which follow the classic minimum XOR topology. Each version has two input neurons (possibly with the inverted pair), two hidden neurons (N1 and N2) and one output neuron (N3). Input and hidden neurons are connected by four synapses and the hidden and the output neuron are connected by a pair of synapses. Weights have been indicated in the synapses and have been achieved for optimal performance of the circuit.

The first type of XOR gate is shown in the figure 7.8. Each neuron in the hidden layer receives the input from one input neuron and from the inverted member of the other input neuron. Thus the hidden neurons either receive 10 or 01. Weights have been set to 3.6 volts so that N1 and N2 will fire if both the inputs are active (that is the input is either 01 or 10) and not if one of the inputs is inactive (when the input is 00 or 11). Thus N1 and N2 become 10 and 01 detectors. Weight for the output neuron is 5 volts so that it will fire if one of the inputs is active. The simulation results are shown in figure 7.9 and 7.10.

In the second scheme (figure 7.11), inverted input neurons are not used. Each of the hidden neurons receives both inputs through excitatory and inhibitory synapses. Weights are set to 4.0 volts so that N1 and N2 can fire if the input is 10 or 01. Synapses for N3 have weights of 5 volts. If the input is 00, there is no activation generated and is totally inhibited if the input is 11. Spice simulation results for inputs 01 and 11 are shown in figure 7.12 and 7.13.

The third scheme is shown in figure 7.14. The weights have been set in such a way that N1 behaves as a logical OR gate and N2 as an AND gate. Output of N2 is fed to N3 by an inhibitory synapse. If the input is 11, both N1 and N2 fire

but because of stronger inhibition at the output, N3 will never fire. Simulations for inputs of 01 and 11 are shown in figure 7.15 and 7.16.

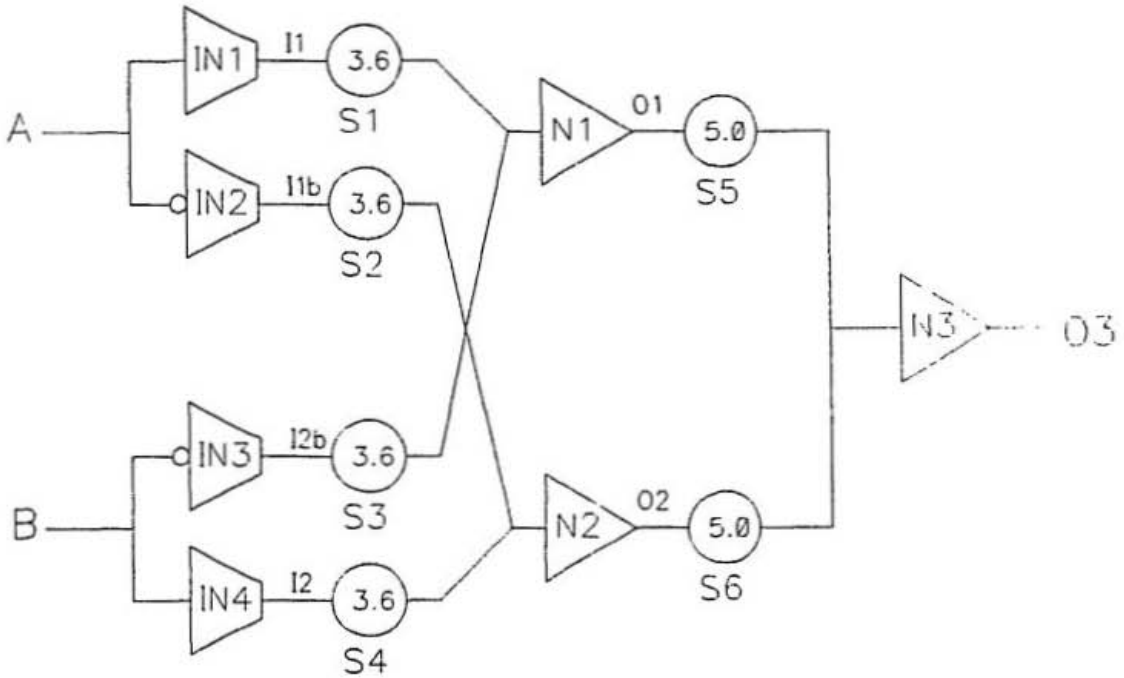

Figure 7.8: Schematic of xor circuit with the weights indicated inside the synapses (represented by circles). Inverting input neurons IN2 and IN3 are used for 0 detection.

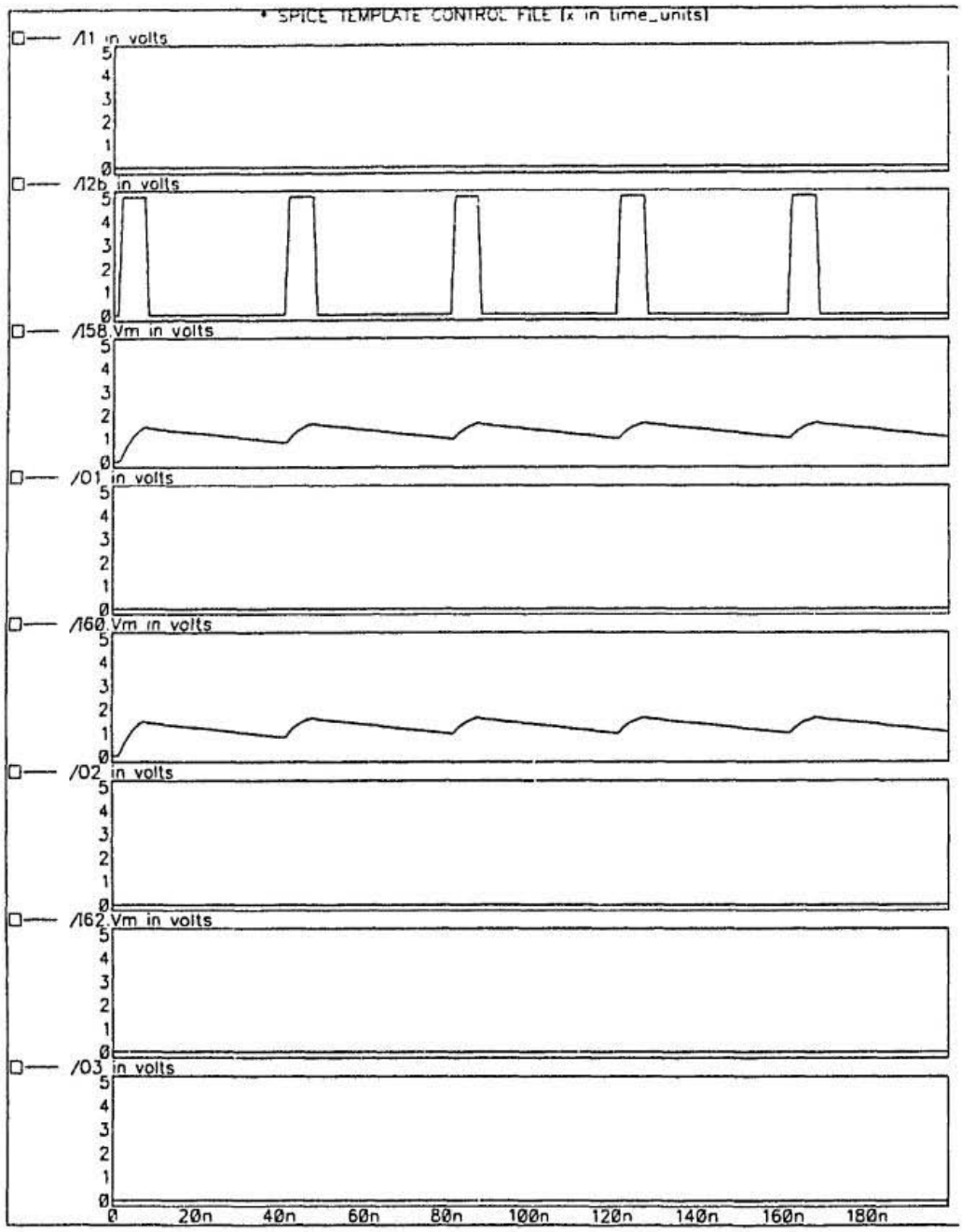

Figure 7.9: Plots for input 00. First two waveforms indicate output of input neurons (one inverted output) and the rest are the activations and outputs of all three neurons.

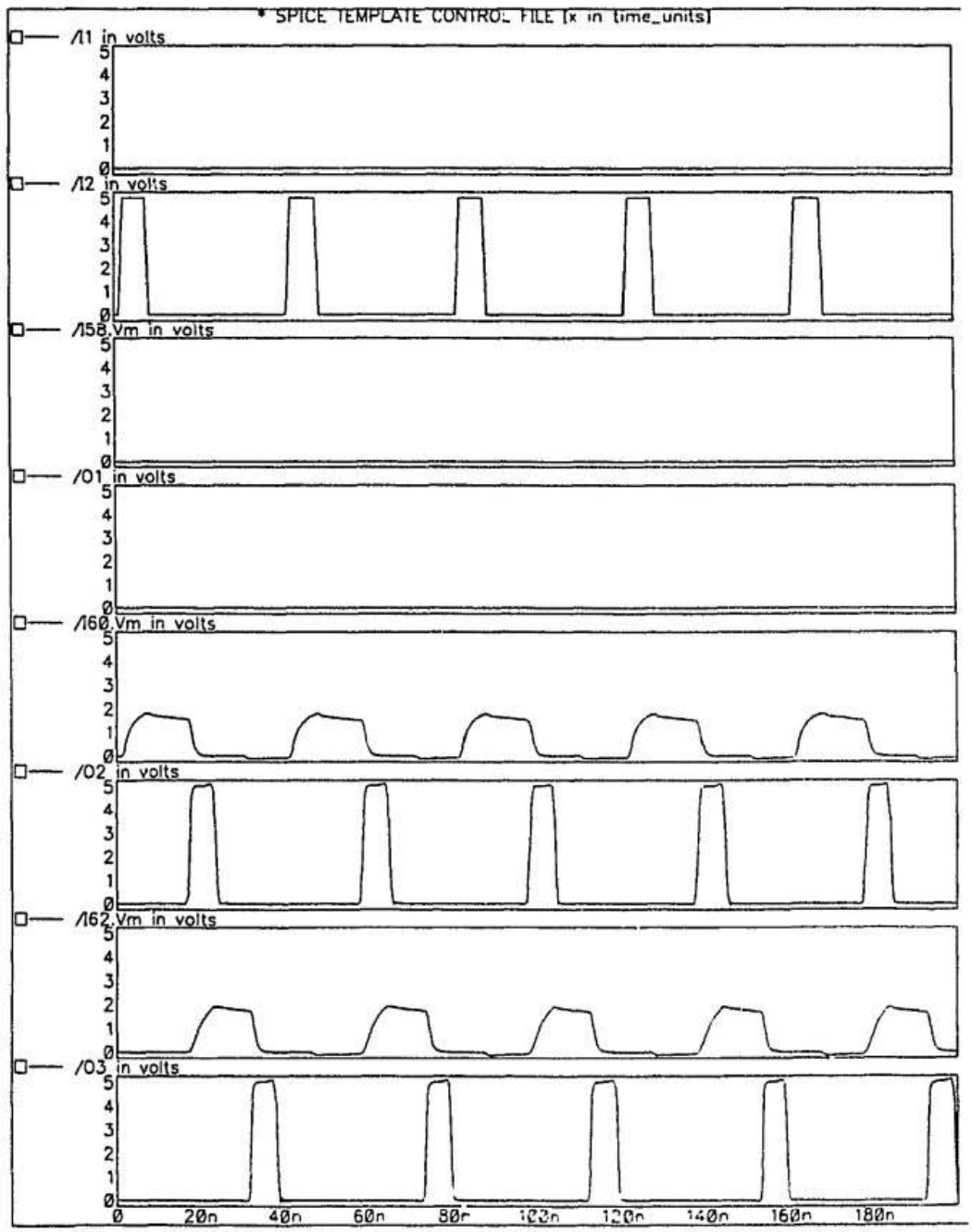

Figure 7.10: Plots for input 01. First two waveforms indicate output of input neurons and the rest are the activations and outputs of all three neurons.

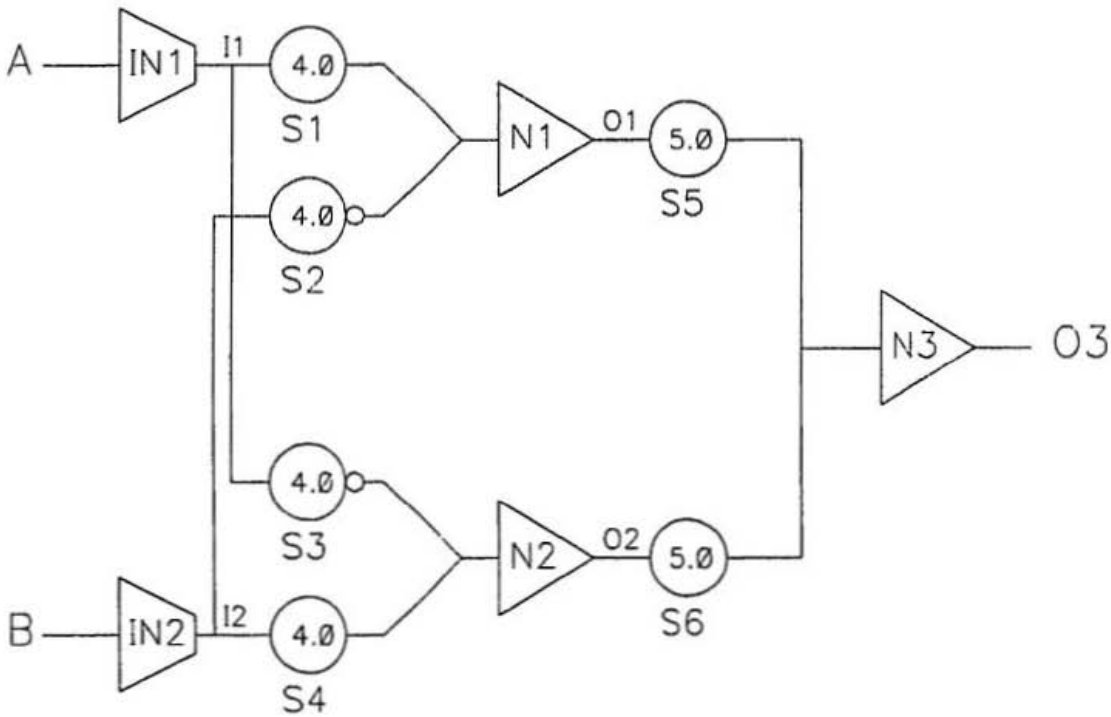

Figure 7.11: Schematic of xor circuit with the weights indicated inside the synapses. Synapse with a small circle in front is inhibitory. N1 and N2 are 10 and 01 detectors.

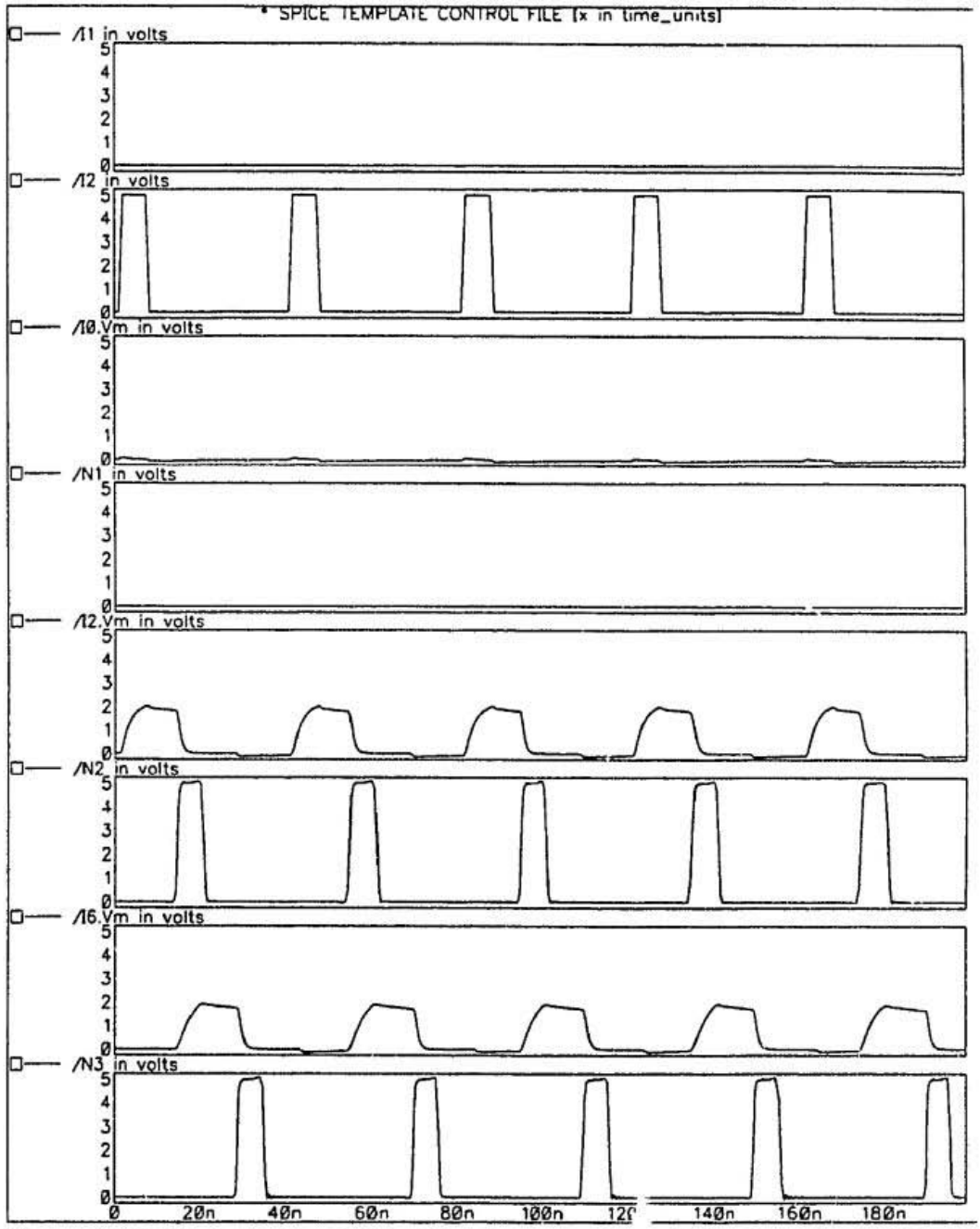

Figure 7.12: Plots for input 01. First two waveforms indicate output of input neurons and the rest are the activations and outputs of all three neurons.

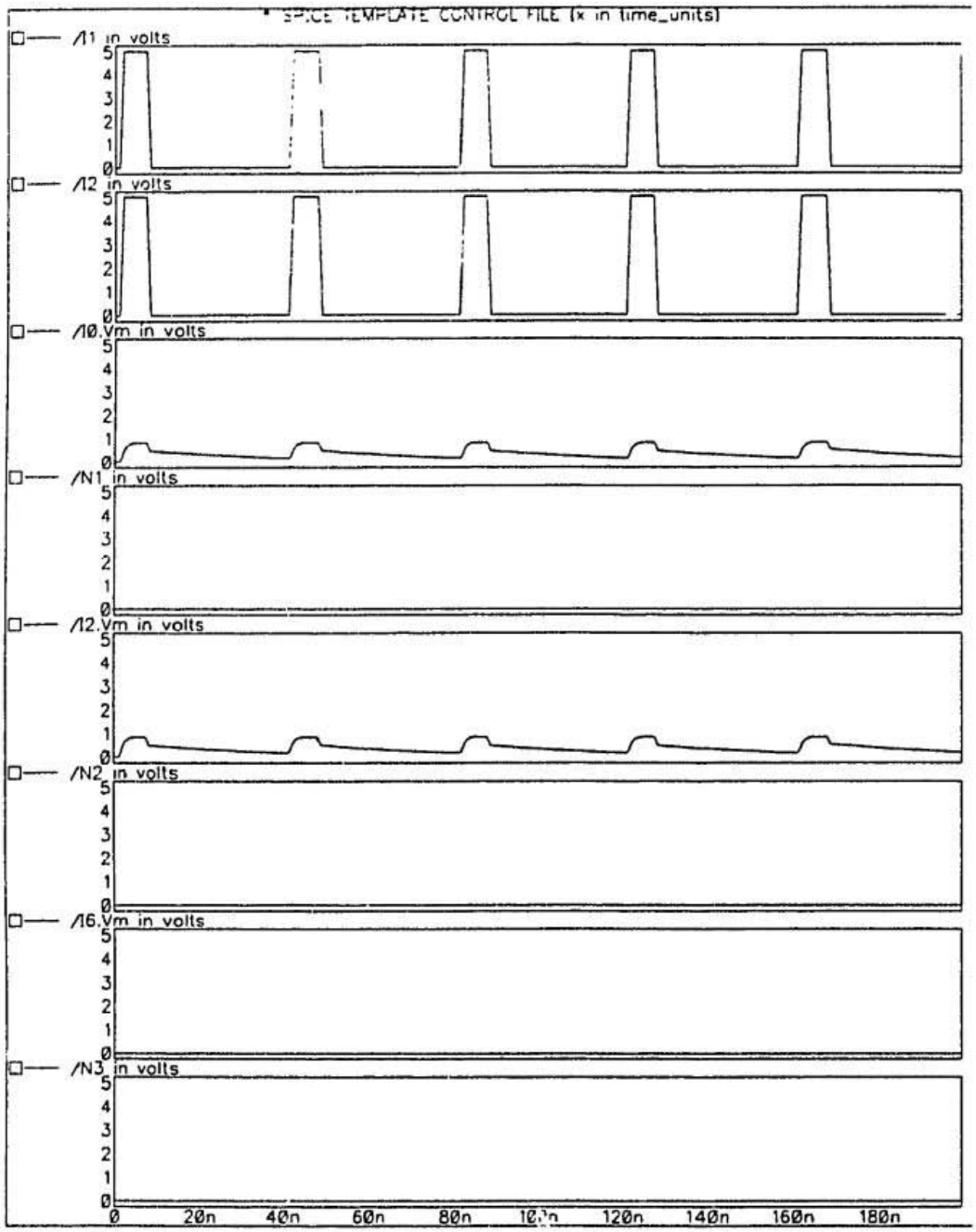

Figure 7.13: Plots for input 11. First two waveforms indicate output of input neurons and the rest are the activations and outputs of all three neurons.

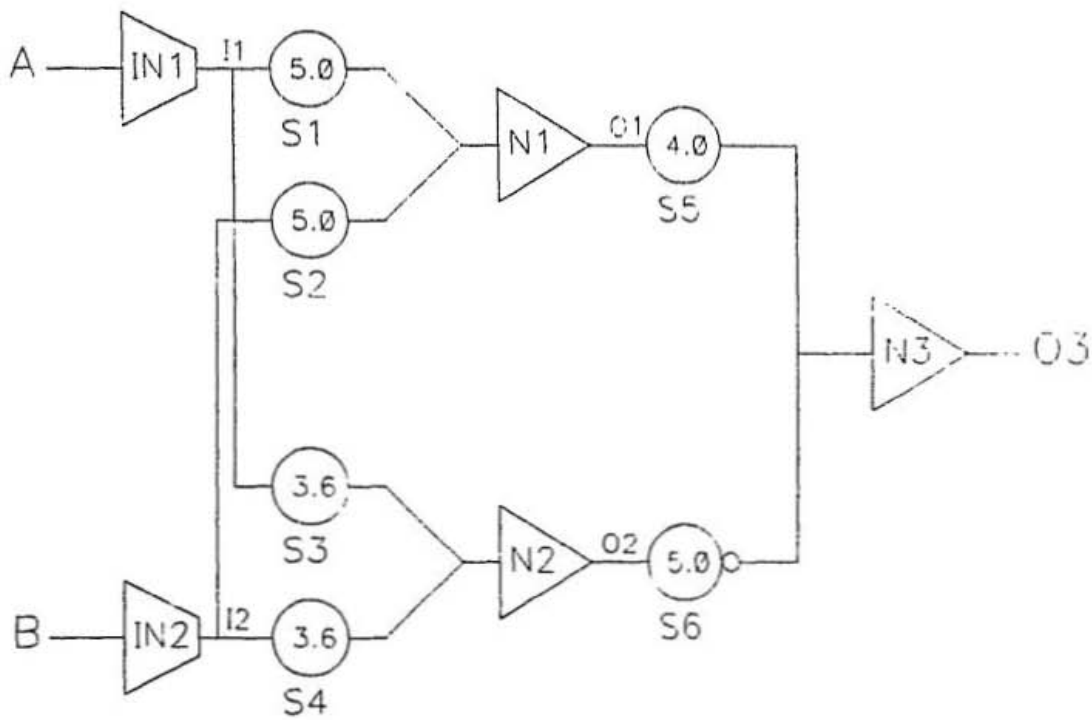

Figure 7.14: Schematic of xor circuit with the weights indicated inside the synapses. S6 is inhibitory synapse. N1 and N2 behave as OR and AND gates.

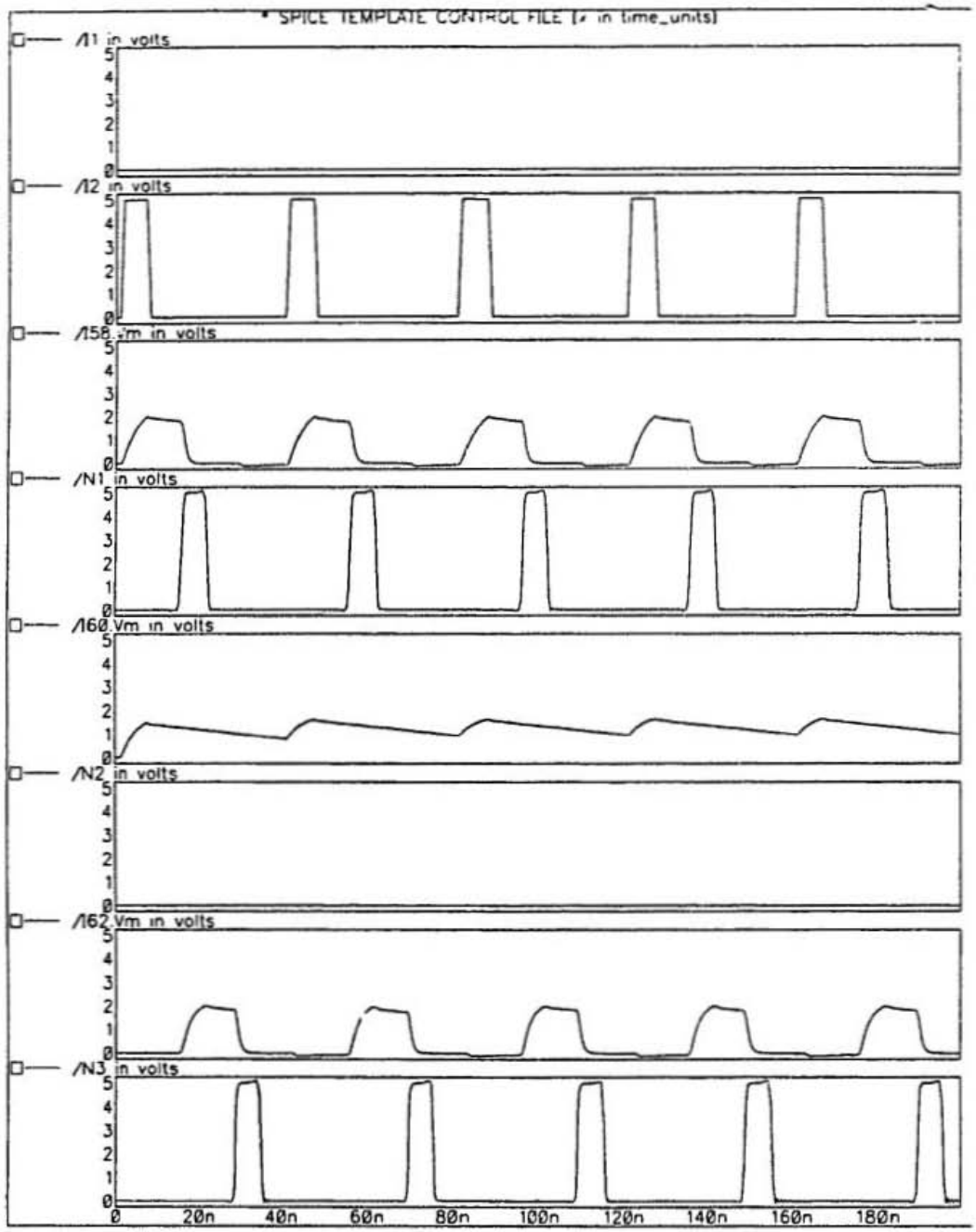

Figure 7.15: Plots for input 01. First two waveforms indicate output of input neurons and the rest are the activations and outputs of all three neurons.

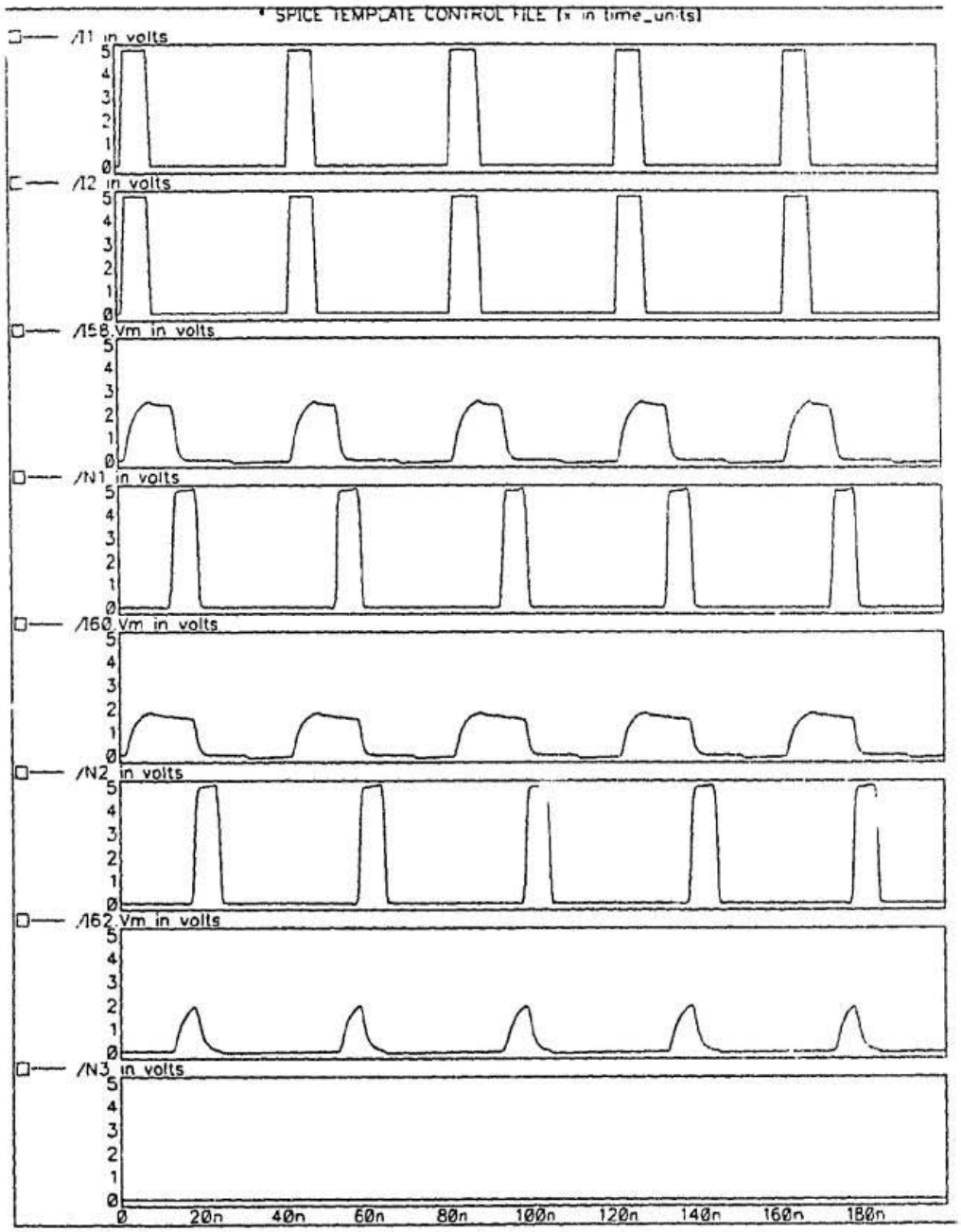

Figure 7.16: Plots for input 11. First two waveforms indicate output of input neurons and the rest are the activations and outputs of all three neurons.

7.3 Cooperative Assignments

In this example [Tank et. al., 87], the network assigns tasks to individuals for optimum performance. One three by three cooperative or task assignment network is shown in figure 7.17. Each neuron in a row represents one individual (X, Y, Z) for a particular task (A, B or C). Performance of X, Y and Z can be encoded in either the input voltage (going to the input neuron) that is the incoming pulse rate, or in the weight. Here the following weights have been generated. Each neuron is connected to all other neurons in the same row and the column by inhibitory synapses. This enables only one neuron to be active in each row and column ensuring that only one individual is assigned to one task. The Hspice simulation is shown in figure 7.18. It can be seen that the network assigned the tasks properly.

	X	Y	Z
A	3.0	4.0	2.5
B	3.5	2.5	3.0
C	3.0	3.0	4.0

Table 7.1: Weight distribution of the 3x3 cooperative assignment net.

7.4 Implementation

Two chips have been designed using $3\mu m$ design rules and will be fabricated through Canadian Microelectronics Corporation. One chip contains all the standard cells and has been laid out manually. This is for testing each cell individually. The other one contains a network similar to figure 7.6 which is a content addressable memory. The outputs of the neurons are gated through AND gates and fed to the inhibitory synapses (figure 7.19). The other inputs of the AND gates are tied together and behave as a control line. If the control line is low, the network behaves like a pattern classifier even though the outputs of excitatory and inhibitory synapses are tied together. The inhibitory synapse does not have any capacitance but the parasitic capacitances (drain to substrate capacitance) come parallel to the total membrane capacitance. Total membrane capacitance with 5 excitatory synapses is 0.75 pF whereas the parasitic capacitance due to 6 inhibitory synapses is less than 0.1 pF. This can be taken care of by increasing the weight voltage. However, when the control line is high, inhibitory synapses are connected and the network works as a content addressable memory. Hspice simulation of the network is given in figure 7.20. The layout has been done using Cadence EDGE auto placement and routing software. Figure 7.21 shows the layout (only metal2 drawing layer) of the network.

7.5 Concluding Remarks

This chapter contains simulation results of some of the standard examples of neural networks. The simulation results show that networks formed by the basic neural componets perform very well. The reason for developing two input neurons has also been discussed. Schematic diagram, Hspice simulation and the layout of one of the chips has also been presented here.

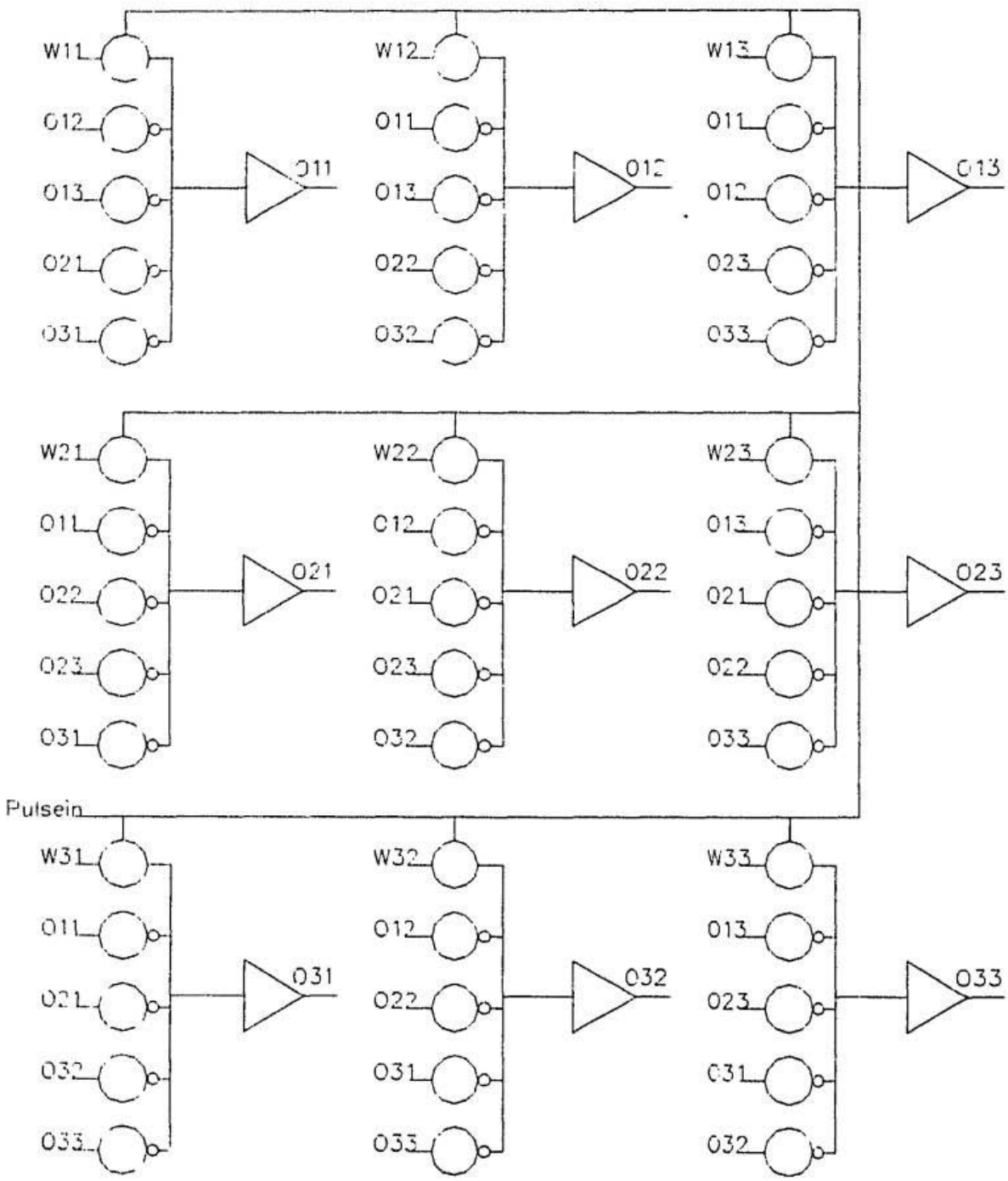

Figure 7.17: 3x3 cooperative assignment network. The excitatory synapse is represented by a circle whereas the inhibitory synapse has a small circle in the front. The excitatory weight is as shown in table 7.1 and the inhibitory weight is 5 volts.

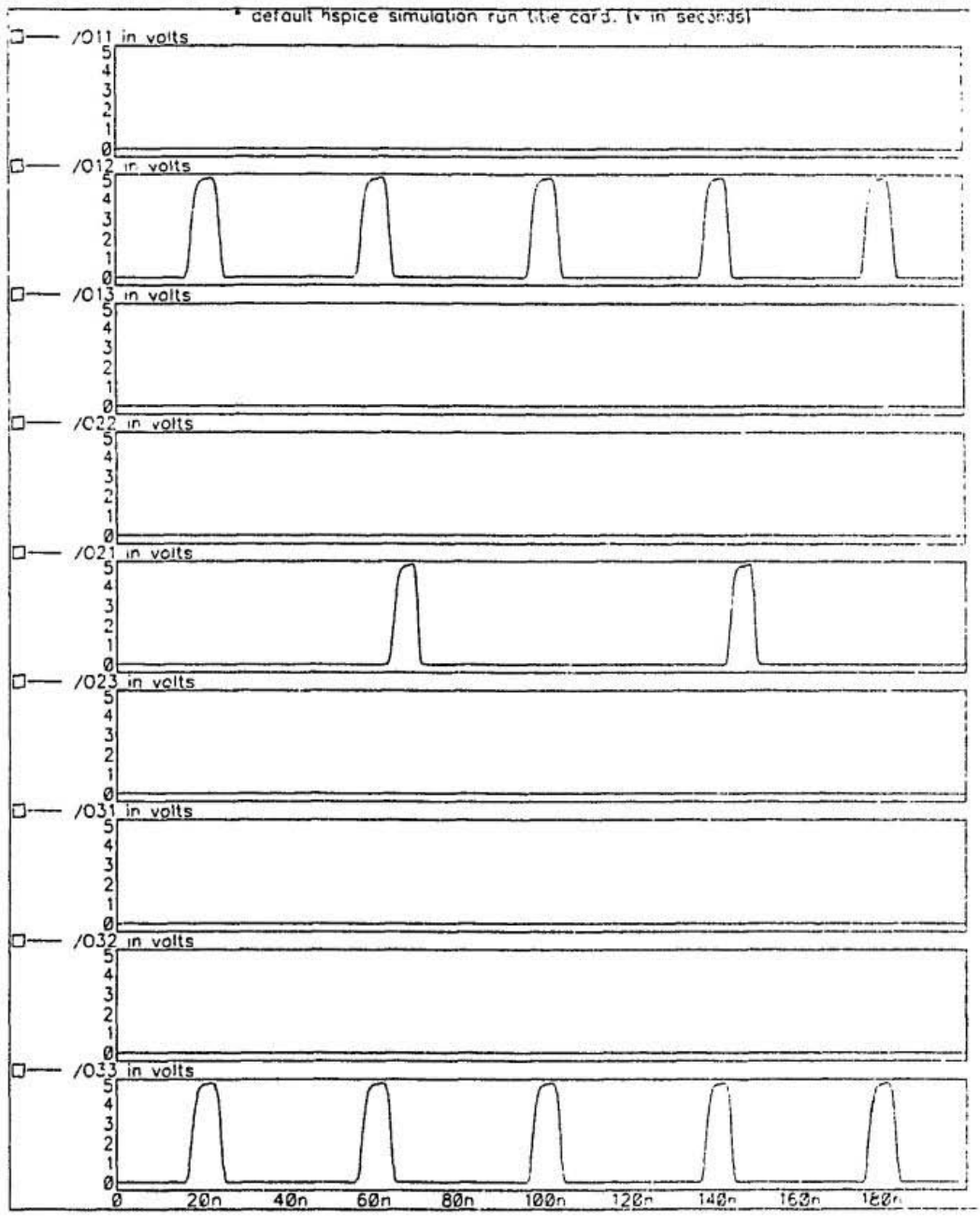

Figure 7.18: Output of all 9 neurons of the 3x3 cooperative assignment network.

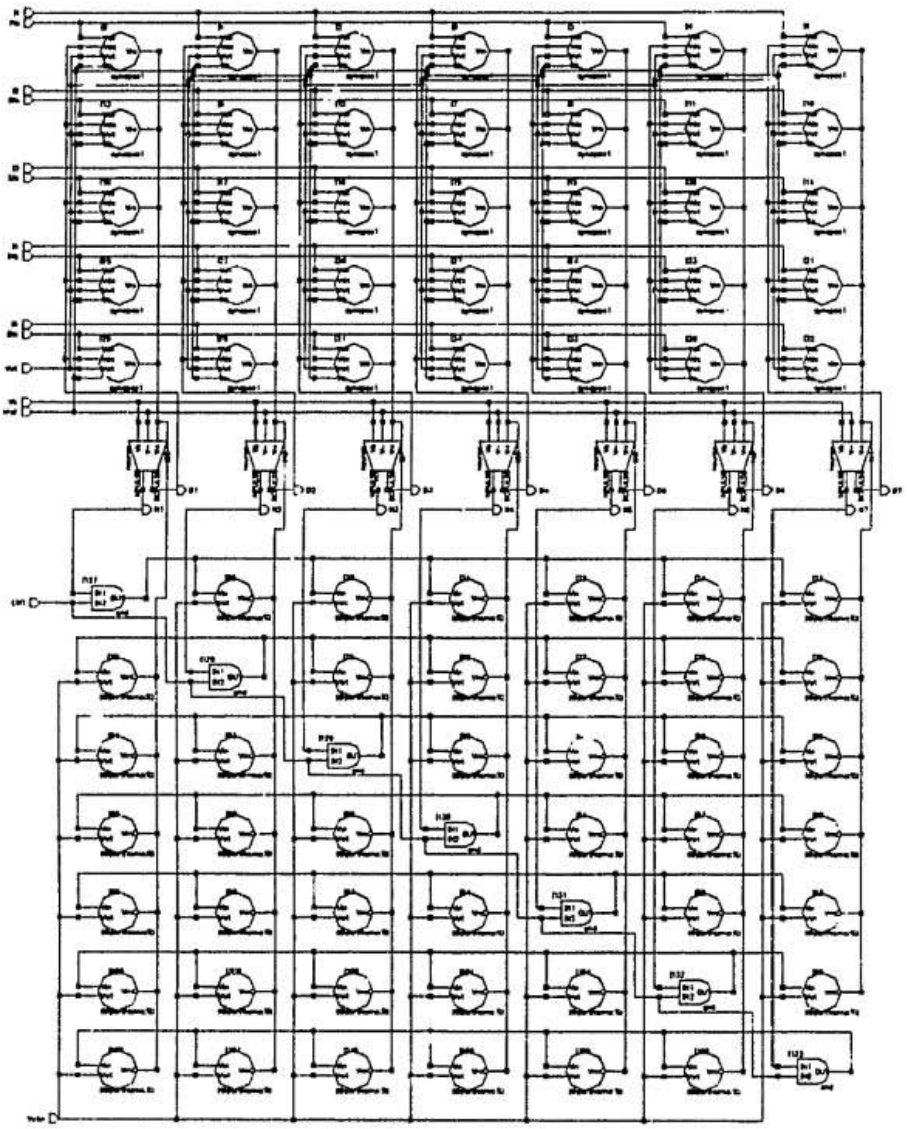

Figure 7.19: Schematic diagram of the controllable pattern classifier / content addressable memory.

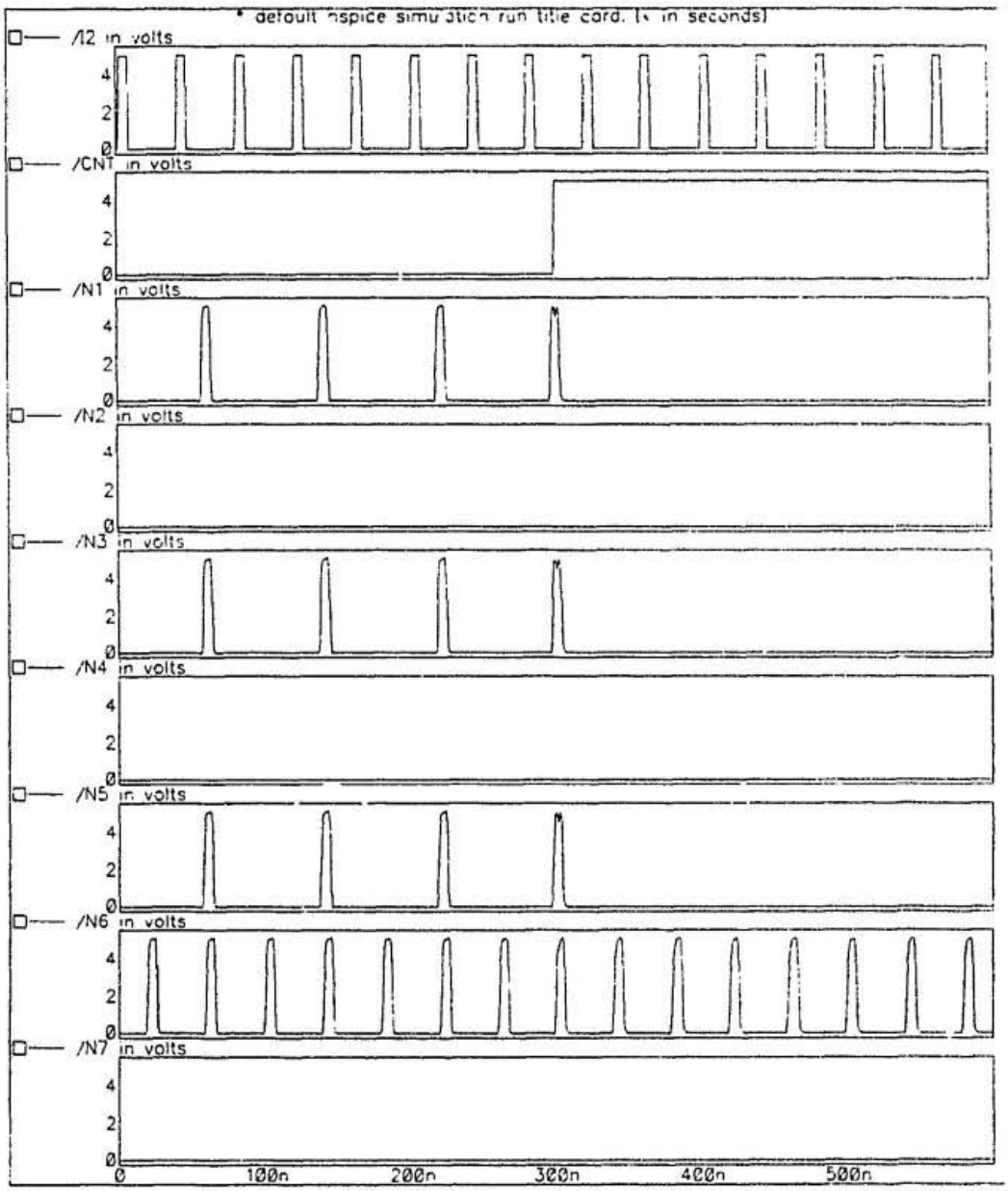

Figure 7.20: Hspice simulation of the CAM. The weight voltage for excitatory synapses is 3.8 volts. The network is probed with 01100 and the waveforms are the control signal and output of all the neurons.

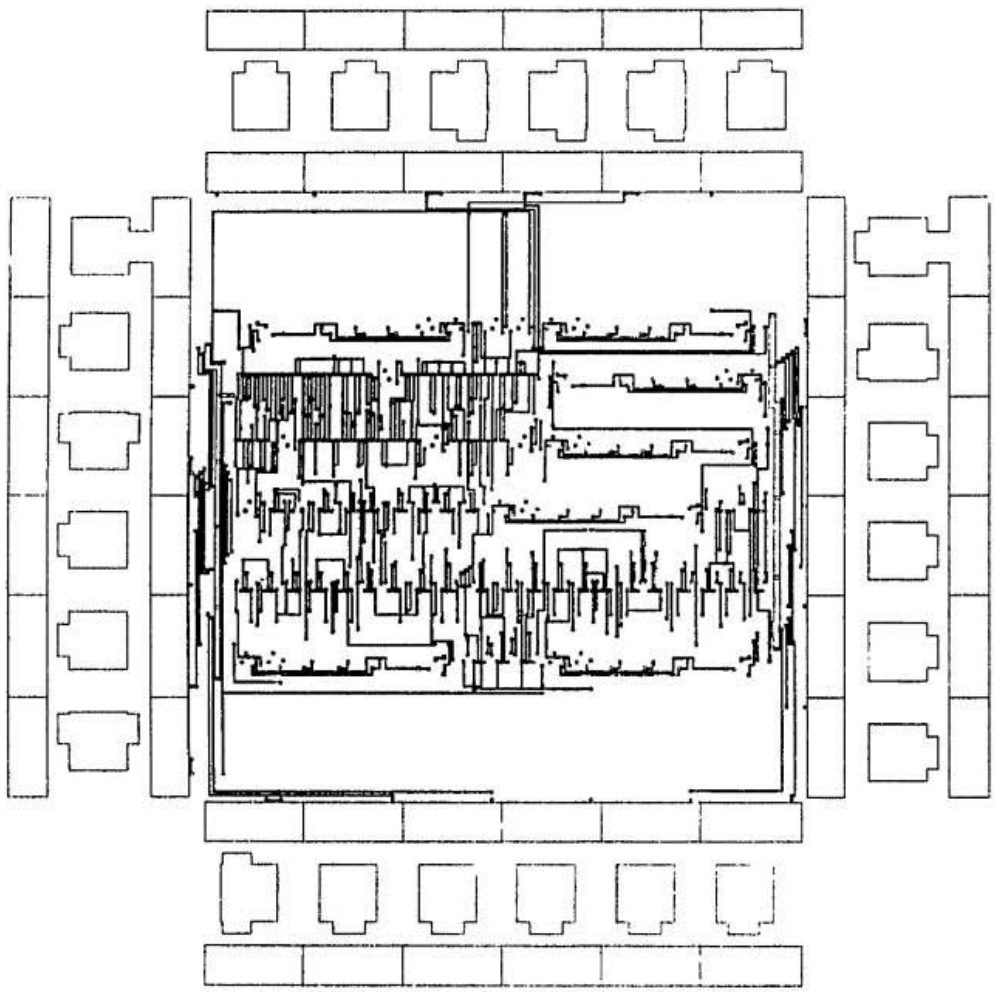

Figure 7.21: Layout of the network (only metal2 drawing layer) obtained by auto place and route routines.

Chapter 8

Conclusions

Pulsed analog neural networks have been described in this thesis where the height of the pulse is modulated by the weight voltage at the gate of an NMOS transistor. The neuron fires one pulse every time the activation exceeds the threshold voltage. At the same time one discharge pulse is also generated to discharge the membrane capacitances. The membrane capacitances and the discharge transistors have been distributed in the synapses allowing the network to be scaled automatically.

The behavior of the networks using the designed circuitry is quite similar to biological neurons though I am not claiming the circuits to be an accurate model of the latter. It deviates from most of the existing neural circuits in a number of aspects. Synapses are very compact enabling one to implement a large network on a chip.

The design procedure for the neural circuits have been given along with a mathematical analysis. Synaptic equations can be used to develop a simulator which can be used for simulating a large network. This is important because it is not possible to simulate a large network in Spice. The output of a small C program using equations 5.13, 5.16 and 5.17 has been given in chapter 5 to check the validity with respect to Spice simulation. The result is very encouraging.

A number of networks have been designed and simulated with Spice. The

results are as expected. Standard cells have been designed and simulation has been done on the extracted schematics. Two chips have been designed using the standard cells and will be fabricated in the Fall of 1991.

A number of interesting points have been observed in the course of the whole work. The relative phase of the incoming pulses have significant effect on the activation voltage of the neuron. Since the neuron has to decide whether to fire a pulse or not in every cycle, this effect of phase difference poses an interesting challenge. Another point is that by varying the membrane capacitance (or effectively τ), the charge integration time is lengthened. This leads to output pulse rates which are integral divisor of the incoming pulse rates. It's effect on a large network is still not known though one can expect the effect to be averaged over a large number of connections. The weighting scheme is non-linear. Whether it has any effect on the learning is still unclear. The only form of learning used so far is to change the weight till proper outputs are obtained. A suitable learning algorithm can also be developed for this kind of networks. If the network can learn properly, then synaptic design in many other systems can be simplified.

References

Aarts E. and Korst J. (1989), "Simulated Annealing and Boltzman Machines", John Wiley and Sons, Chapter 7, pp 117-128.

Allen P. and Holberg D. (1987), "CMOS Analog Circuit Design", Rinehart and Winston.

Blayo F. and Hurat P. (1989), "A VLSI Systolic Array Dedicated to Hopfield Neural Network", *VLSI for Artificial Intelligence*, Edited by J. G. Delgado-Frias & W. R. Moore, Kluwer Academic Publishers, pp 255-264.

Brownlow M. J., Tarassenko L. and Murray A. F. (1990), "Analogue Computation Using VLSI Neural Networks Devices", *Electronics Letter*, Vol. 26, No. 16, pp 1297-1299.

Butler Z., Murray Alan and Smith Antony (1989), "VLSI Bit-Serial Neural networks", *VLSI for Artificial Intelligence*, Edited by J. G. Delgado-Frias & W. R. Moore, Kluwer Academic Publishers, pp 201-208.

CMOS3 DLM Cell Library (1989), Report ICI-020R0, Canadian Microelectronics Corporation.

Cotter Neil E., Smith Kent and Gasper Martin (1988), "A Pulse-width Modulation Design Approach and Path Programmable Logic for Artificial Neural Networks", *Advanced Research in VLSI* ,Proceedings of the Fifth, MIT Conference, March, pp 1-18.

Eberhardt S., Duong T. and Thakoor A. (1989), "Design of a Parallel Hardware Neural Network System from Custom Analog VLSI Building Block Chips", IJCNN, IEEE, Piscataway, NJ. pp 183-190.

El-Leithy N., Newcomb R. W. and Zaghloul M. (1987), "A Basic MOS Neural-type Junction", *IEEE International Conference on Neural Networks*, vol III, pp 469-477.

Faure B. and Mazare G. (1989), "A VLSI Implementation of Multilayered Neural Networks", *VLSI for Artificial Intelligence*, Edited by J. G. Delgado-Frias & W. R. Moore, Kluwer Academic Publishers, pp 159-168.

Geiger R., Allen P. and Strader N. (1990), "VLSI Design Techniques for Analog and Digital Circuits", McGraw Hill Publishing Company.

Graf H. P., Jackel L. D., Howard R. E., Straughn B., Denker J. S., Hubbard W., Tennant D. M. and Schwartz D. (1986), "VLSI Implementation of a Neural Network Memory with Several Hundreds of Neurons", *AIP Conference Proceedings 151, Neural Networks for Computing*, Snowbird, Utah, pp 182-187.

Graf H. P. and deVegvar P. (1987), "A CMOS Implementation of a Neural Network Model" *Advanced Research in VLSI*. Proceedings of the 1987 Stanford Conference, P. Losleben, editor, the MIT Press, pp 351-367.

Graf H. P., Jackel L. D. and Hubbard W. E. (1988), "VLSI Implementation of a Neural Network Model", *IEEE Computer*, March, pp 41-49.

Graf H. P., Jackel L. D. (1989), "Analog Electronic Neural network Circuits", *IEEE Circuits and Devices Magazine*, July, pp 44-49,55.

Hirai Y., Kamada K., Yamada M. and Ooyama M. (1989), "A Digital Neuro-chip with Unlimited Connectability for Large Scale Neural Network", IJCNN, IEEE, Piscataway, NJ, vol II, pp 163-169.

Hollis P. and Paulos J. (1990), "Artificial Neural Networks using MOS Analog Multipliers", *IEEE Journal of Solid-State Circuits*, Vol. 25, No. 3, pp 849-855.

Hopfield J. J. (1982), "Neural Networks and Physical Systems with Emergent Collective Computational Abilities", *Proc. Natl. Acad. Sc. USA*, Vol 79, pp 2254-2258.

Hopfield J. J. (1984), "Neurons with Graded Response Have Collective Computational Properties Like Those of Two-state Neurons", *Proc. Natl. Acad. Sc. USA*, Vol 81, pp 3088-3092.

Hubbard W., Schwartz D., Denker J., Graf H., Howard R., Jackel L., Straughn B. and Tennant D. (1986), "Electronic Neural Networks", *AIP Conference Proceedings 151, Neural Networks for Computing*, Snowbird, Ut, pp 227-234.

Koester J. (1981), "Resting Membrane Potential", Chapter 3, *Principles of Neural Science*, Kandel E. R. & Schwartz J. H. (1984), editors, Publisher : Elsevier/North-Holland, pp 27-35.

Koester J. (1981A), "Active Conductances Underlying the Action Potentials", Chapter 6, *Principles of Neural Science*, Kandel E. R. & Schwartz J. H. (1984), editors, Publisher : Elsevier/North-Holland, pp 53-62.

Lauther U. (1983), "Cell Based VLSI Design System", *Hardware and Software Concepts in VLSI*, edited by G. Rabbat, Van Nostrand Reinhold Company, pp 480-494.

Mead C. (1989), "Analog VLSI and Neural Systems", Addision-Wesley Publishing Company.

Mueller P., Van der Spiegel J., Blackman D., Chiu T., Clare T., Donham J. C., Hsieh T. and Loinaz M. (1989), "A General Purpose Analog Neural Computer", IJCNN, IEEE, Piscataway, NJ, vol II, pp 177-182.

Mukherjee A. (1985), "Introduction to NMOS and CMOS VLSI Systems Design", Prentice Hall.

Murray A. F., Hamilton A. and Tarassenko L. (1989), "Programmable Analog Pulse Firing Neural networks", *Advances in Neural Information Processing Systems*, Edited by D. S. Touretzky, Morgan Kaufmann Publishers, pp 671-677.

Murray A. F., Del Corso D. and Tarassenko L. (1991), "Pulse-Stream VLSI Neural Networks Mixing Analog and Digital Techniques", *IEEE Transactions on Neural Networks*, Vol. 2, No. 2, pp 193-204.

Personnaz L., Guyon I., Johannet A., Dreyfus G. and Toulouse G. (1986), "A Simple Selectionist Learning Rule for Neural Networks", *AIP Conference*

Proceedings 151. Neural Networks for Computing, Snowbird, Ut, pp 360-363.

Rumelhart D. E., McClelland J. L. and The PDP Research Group (1984), *Parallel Distribution Processing, Explorations in the Microstructure of Cognition*, Volume 1, the MIT Press.

Sage J. P., Thompson K., Withers R. S. (1986), "Artificial Neural Network Integrated Circuit Based on MNOS/CCD Principles", *AIP Conference Proceedings 151. Neural Networks for Computing*, Snowbird, Ut, pp 381-385.

Schwartz D., Howard R. and Hubbard W. (1989), "Adaptive Neural networks using MOS Charge Storage", *Advances in Neural Information Processing Systems*, Edited by D. S. Touretzky, Morgan Kaufmann Publishers, pp 761-768.

Spencer E. G. (1986), "Programmable Bistable Switches and resistors for Neural Networks", *AIP Conference Proceedings 151. Neural Networks for Computing*, Snowbird, Ut, pp 414-419.

Sze S. M. (1981), "Physics of Semiconductor Devices", 2nd edition, A Wiley-Interscience Publication.

Tank D. W. and Hopfield J. (1986), "Simple Neural Optimization Networks", *IEEE Transaction on Circuits and Systems*, Vol CAS 33, No. 5, pp 533-554.

Tank D. W. and Hopfield J. (1987), "Collective Computation in Neuronlike Circuits", *Scientific America*, Special Issue, Vol 1.

Tomberg J. and Kaski K. (1990), "Pulse Density Modulation Technique in VLSI Implementations of Neural Network Algorithms", *IEEE Journal of Solid-State Circuits*, Vol. 25, No. 5, pp 1277-1290.

Verleysen M., Sieletti B. and Jespers P. (1989), "A New CMOS Architecture for Neural Networks", *VLSI for Artificial Intelligence*, Edited by J. G. Delgado-Frias & W. R. Moore, Kluwer Academic Publishers, pp 209-217.

Videoconferences, IEEE, (1991), *Neural Network Applications for the 1990's*, The 42nd Videoconferences via Satellite, May 23.

Vladimirescu A. and Liu S. (1980), "The Simulation of MOS Integrated Circuits using SPICE2", Memo no. UCB/ERL M80/7, College of Engineering, University of California, Berkeley.

Weinfield Michel (1989), "A fully Digital Integrated Hopfield Network Including the Learning Algorithm", *VLSI for Artificial Intelligence*, Edited by J. G. Delgado-Frias & W. R. Moore, Kluwer Academic Publishers, pp 169-178.

Appendix

This is a typical input deck for Spice created by Cadence Edge 2.1. This is the input deck for simulation of standard neuron (fig 5.8 and 5.9).

* net 1 = vdd!
* net 0 = gnd!
* net 2 = /Vb
* net 3 = /V-
* net 4 = /V+
* net 5 = /I53.OUT
* net 6 = /I46.OUT
* net 7 = /I47.OUT
* net 8 = /I57.OUT
* net 9 = /DCPULSE
* net 10 = /I31.OUT
* net 11 = /I60.OUT
* net 12 = /I52.OUT
* net 13 = /I37.B
* net 14 = /NPULSE
* net 15 = /I23.Vout
* net 18 = /I60/I0.D

```
.MODEL Model4 nmos level=3 vto=.7 kp=4.e-05 gamma=1.1 phi=.6
+lambda=.01 pb=.7 cgso=3.e-10 cgdo=3.e-10 cgbo=5.e-10 rsh=25
+cj=.00044 mj=.5 cjsw=4.e-10 mjsw=.3 js=1.e-05 tox=5.e-08
+nsub=1.7e+16 nss=0 nfs=0 tpg=1 xj=6.e-07 ld=3.5e-07 uo=775
+utra=0 vmax=1.e+05 xqc=.5 theta=.13 eta=.05 kappa=1
nmos(4) = /I60/I4
M$#4 11 18 0 0 Model4 l=3u w=5.4u
nmos(5) = /I60/I0
M$#5 18 10 0 0 Model4 l=3u w=5.4u
.MODEL Model5 pmos level=3 vto=-.8 kp=1.2e-05 gamma=.6 phi=.6
+lambda=.03 pb=.6 cgso=2.5e-10 cgdo=2.5e-10 cgbo=5.e-10 rsh=80
+cj=.00015 mj=.6 cjsw=4.e-10 mjsw=.6 js=1.e-05 tox=5.e-08
+nsub=5.e+15 nss=0 nfs=0 tpg=1 xj=5.e-07 ld=2.5e-07 uo=250
+utra=0 vmax=70000 xqc=.5 theta=.13 eta=.3 kappa=1
pmos(6) = /I60/I7
M$#6 1 18 11 1 Model5 l=3u w=5.4u
pmos(7) = /I60/I1
M$#7 1 10 18 1 Model5 l=3u w=5.4u
net 21 = /I57/I0.D
nmos(12) = /I57/I4
M$#12 8 21 0 0 Model4 l=3u w=5.4u
nmos(13) = /I57/I0
M$#13 21 5 0 0 Model4 l=3u w=5.4u
pmos(14) = /I57/I7
M$#14 1 21 8 1 Model5 l=3u w=5.4u
pmos(15) = /I57/I1
```

M$#15 1 5 21 1 Model5 l=3u w=5.4u

nmos(16) = /I49/I0

M$#16 14 7 0 0 Model4 l=3u w=5.4u

pmos(17) = /I49/I1

M$#17 1 7 14 1 Model5 l=3u w=10.8u

nmos(20) = /I48/I0

M$#20 9 6 0 0 Model4 l=3u w=5.4u

pmos(21) = /I48/I1

M$#21 1 6 9 1 Model5 l=3u w=10.8u

net 29 = /I47/I1.D

nmos(26) = /I47/I1

M$#26 29 11 0 0 Model4 l=3u w=10.8u

nmos(27) = /I47/I3

M$#27 7 13 29 0 Model4 l=3u w=10.8u

pmos(28) = /I47/I2

M$#28 1 11 7 1 Model5 l=3u w=10.8u

pmos(29) = /I47/I0

M$#29 1 13 7 1 Model5 l=3u w=10.8u

net 33 = /I46/I1.D

nmos(32) = /I46/I1

M$#32 33 13 0 0 Model4 l=3u w=10.8u

nmos(33) = /I46/I3

M$#33 6 8 33 0 Model4 l=3u w=10.8u

pmos(34) = /I46/I2

M$#34 1 13 6 1 Model5 l=3u w=10.8u

pmos(35) = /I46/I0

M$#35 1 8 6 1 Model5 l=3u w=10.8u

capacitor(36) = /I59/I0

C$#36 13 0 poly .1pf

capacitor(38) = /I58/I0

C$#38 12 0 poly .1pf

capacitor(40) = /I54/I0

C$#40 5 0 poly .1pf

capacitor(42) = /I43/I0

C$#42 11 0 poly .1pf

net 38 = /I37/I10.OUT

nmos(44) = /I37/I12/I0

M$#44 13 38 0 0 Model4 l=3u w=5.4u

pmos(45) = /I37/I12/I1

M$#45 1 38 13 1 Model5 l=3u w=10.8u

nmos(48) = /I37/I10/I0

M$#48 38 15 0 0 Model4 l=3u w=5.4u

pmos(49) = /I37/I10/I1

M$#49 1 15 38 1 Model5 l=3u w=10.8u

net 47 = /I53/I0.D

net 48 = /I53/I4.D

nmos(58) = /I53/I11

M$#58 5 48 0 0 Model4 l=3u w=5.4u

nmos(59) = /I53/I4

M$#59 48 47 0 0 Model4 l=3u w=5.4u

nmos(60) = /I53/I0

M$#60 47 12 0 0 Model4 l=3u w=5.4u

pmos(61) = /I53/I10

M$#61 1 48 5 1 Model5 l=3u w=5.4u

pmos(62) = /I53/I7

M$#62 1 47 48 1 Model5 l=3u w=5.4u

pmos(63) = /I53/I1

M$#63 1 12 47 1 Model5 l=3u w=5.4u

net 51 = /I52/I0.D

net 52 = /I52/I4.D

nmos(70) = /I52/I11

M$#70 12 52 0 0 Model4 l=3u w=5.4u

nmos(71) = /I52/I4

M$#71 52 51 0 0 Model4 l=3u w=5.4u

nmos(72) = /I52/I0

M$#72 51 11 0 0 Model4 l=3u w=5.4u

pmos(73) = /I52/I10

M$#73 1 52 12 1 Model5 l=3u w=5.4u

pmos(74) = /I52/I7

M$#74 1 51 52 1 Model5 l=3u w=5.4u

pmos(75) = /I52/I1

M$#75 1 11 51 1 Model5 l=3u w=5.4u

net 55 = /I31/I0.D

net 56 = /I31/I4.D

nmos(82) = /I31/I11

M$#82 10 56 0 0 Model4 l=3u w=5.4u

nmos(83) = /I31/I4

M$#83 56 55 0 0 Model4 l=3u w=5.4u

nmos(84) = /I31/I0

M$#84 55 13 0 0 Model4 l=3u w=5.4u

pmos(85) = /I31/I10

M$#85 1 56 10 1 Model5 l=3u w=5.4u

pmos(86) = /I31/I7

M$#86 1 55 56 1 Model5 l=3u w=5.4u

pmos(87) = /I31/I1

M$#87 1 13 55 1 Model5 l=3u w=5.4u

net 61 = /I23/I0.S

net 62 = /I23/I0.D

net 63 = /I23/I1.D

pmos(90) = /I23/I6

M$#90 1 63 15 1 Model5 l=3u w=5.4u

pmos(91) = /I23/I2

M$#91 1 62 63 1 Model5 l=3u w=5.4u

pmos(92) = /I23/I3

M$#92 1 62 62 1 Model5 l=3u w=5.4u

nmos(93) = /I23/I4

M$#93 61 2 0 0 Model4 l=3u w=10.8u

nmos(94) = /I23/I5

M$#94 15 2 0 0 Model4 l=6u w=5.4u

nmos(95) = /I23/I1

M$#95 63 4 61 0 Model4 l=3u w=5.4u

nmos(96) = /I23/I0

M$#96 62 3 61 0 Model4 l=3u w=5.4u